



ur72

APLICACIÓN WEB PARA LA PREDICCIÓN DEL RETORNO
DE PACIENTES AL SERVICIO DE URGENCIAS DE UN
HOSPITAL EN EL PLAZO DE 72 HORAS

WEB APP FOR RETURN PREDICTION OF PATIENTS
AT THE EMERGENCY DEPARTMENT OF A
HOSPITAL WITHIN 72 HOURS

TRABAJO DE FIN DE GRADO DEL
GRADO EN INGENIERÍA DE COMPUTADORES
FACULTAD DE INFORMÁTICA

AUTOR
JUAN PEDRO OLMO ROJAS

PROFESOR TUTOR
ANTONIO SARASA CABEZUELO

CURSO ACADÉMICO 2020/2021



UNIVERSIDAD
COMPLUTENSE
MADRID

AGRADECIMIENTOS

Al hacer una primera reflexión y echar la vista atrás, lo primero que viene a mi mente es un grupo de personas que hace años comenzó a estudiar informática. En el aula 1210 del edificio multiusos y en horario de tarde se encontraron. Eran de diferentes países, diferentes edades y diferentes culturas, pero les unió un mismo propósito, ser ingenieros informáticos.

Gracias a ellos hoy me encuentro aquí, sin su ayuda no hubiera sido posible, por tanto, mi primer y más cariñoso agradecimiento es para ellos, mis compañeros de batalla.

Después me gustaría reconocer la labor a esos profesores que creyeron en mí y me apoyaron, a veces un pequeño gesto ayuda más que una buena lección.

Para mi familia y amigos también hay agradecimiento, por acompañarme y soportarme en estos años. Espero dedicaros más tiempo a partir de ahora.

Por último, quiero expresar mi gratitud a mi tutor Antonio Sarasa, con el que repito después de haberme impartido clase años atrás. En primer lugar, gracias por la oportunidad de realizar este trabajo con él. En segundo lugar, gracias por su disponibilidad y flexibilidad a la hora de guiarme en este proyecto.

ÍNDICE GENERAL

RESUMEN	4
PALABRAS CLAVE.....	4
ABSTRACT	4
KEYWORDS	4
1 INTRODUCCIÓN	5
1.1 MOTIVACIÓN.....	5
1.2 OBJETIVOS	6
1.3 ESTADO DEL ARTE	6
1.4 PLAN DE TRABAJO	7
2 INTRODUCTION	8
3 ESPECIFICACIÓN DE REQUISITOS.	9
3.1 ACTORES.....	9
3.2 MÓDULOS.....	9
3.2.1 MÓDULO ADMINISTRADOR	9
3.2.2 MÓDULO USUARIO	10
3.3 CASOS DE USO	10
4 TECNOLOGÍAS EMPLEADAS.....	17
4.1 APLICACIÓN WEB	17
4.2 RED NEURONAL.....	18
4.3 BASE DE DATOS	18
5 ARQUITECTURA DE LA APLICACIÓN	19
6 MODELO DE DATOS.....	20
6.1 Tabla “asistencias”	20
6.2 Tabla “usuarios”	22
6.3 Tabla “rol”	23
6.4 Tabla “asis_test”	23
6.5 Relaciones entre tablas	24
7 DISEÑO	25
7.1 Página inicial de registro	25
7.2 Página principal	26
8 IMPLEMENTACIÓN	27
8.1 IMPLEMENTACIÓN RED NEURONAL	27
8.2 IMPLEMENTACIÓN APLICACIÓN WEB.....	30
9 EVALUACIÓN	36

10	CONCLUSIONES Y TRABAJO FUTURO	37
10.1	CONCLUSIONES	37
10.2	CONCLUSIONS	38
10.3	TRABAJO FUTURO	39
11	GUÍA DE USO	40
11.1	INICIO	40
11.2	IMPORTAR DATOS	43
11.3	HISTORIALES	44
11.4	USUARIOS	44
11.5	PREDICCIONES ur72 MANUAL	46
11.6	PREDICCIONES ur72 HISTÓRICO	47
11.7	GRÁFICAS	48
11.7.1	GRAFICAS 1	48
11.7.2	GRÁFICAS 2	49
11.7.3	GRAFICAS 3	50
11.7.4	GUARDAR PDF	51
11.8	SELECCIONAR DATOS	52
12	REPOSITORIO DE CÓDIGO	52
13	BIBLIOGRAFÍA	53
13.1	REFERENCIAS	53

RESUMEN

Uno de los factores que determina la calidad de los servicios de urgencias de cualquier hospital, es la tasa de retorno de los pacientes a dichos servicios en un corto plazo de tiempo (24, 48 o 72 horas).

En este trabajo se pretende crear una herramienta que ayude a los profesionales de los centros hospitalarios en su labor. Para ello se ha creado una aplicación web, donde una de las principales funciones será pronosticar el porcentaje que tiene un paciente de regresar al hospital en un periodo de 72 horas, si es dado de alta.

Como punto de partida disponemos de los informes de ingresos de pacientes del “Hospital San Juan de Dios del Aljarafe”, Sevilla, de los años 2015, 2016, 2017 y 2018. Con estos datos alimentaremos una red neural que se encargará de realizar la predicción.

PALABRAS CLAVE

Red neuronal, python, hospital, urgencias, web, JavaScript, php, MySQL.

ABSTRACT

One of the factors that determines the quality of emergency services in any hospital is the rate return of patients to these services in a short period of time (24, 48 or 72 hours).

The target of this work, is to create a tool that helps hospital professionals in their work. For it, this web application has been created, one of the main functions will be to forecast the percentage that a patient has to return to the hospital in a period of 72 hours, if he is discharged.

As a starting point, we have the patient admissions reports from the “Hospital San Juan de Dios del Aljarafe”, Sevilla, of the years 2015, 2016, 2017 and 2018. With this data we will feed a neural network that will be in charge of making the prediction.

KEYWORDS

Neuronal Network, python, hospital, emergencies, web, JavaScript, php, MySQL.

1 INTRODUCCIÓN

En el presente donde nos encontramos, en medio de una pandemia mundial, nos damos cuenta que la gestión de los recursos sanitarios es de vital importancia.

Por desgracia, nos ha tocado ver por televisión e incluso sufrir en primera persona el colapso de numerosos centros hospitalarios.

Ante este desbordamiento, podríamos plantear el debate de quién tiene prioridad para ser atendido cuando no hay medios suficientes para todos, ¿primero los jóvenes y después los ancianos? Sería un debate interminable, cada cual tendría su opinión, unos desde el punto de vista social, otros desde el religioso, el moral, político y tantos otros. Todos válidos, por qué no.

¿Quién debe tener la potestad para mediar en estas situaciones?

1.1 MOTIVACIÓN

En mi opinión, si se me permite, la decisión debe ser tomada por los profesionales de este sector, es decir, los médicos. Personas que han dedicado mucho tiempo y esfuerzo en formarse para salvar vidas.

Como digo, son personas y también son humanas, ayudémosles entonces. De esta premisa surge la herramienta *ur72*, poner a disposición del personal sanitario una aplicación web donde poder obtener una predicción basada en registros reales. Y no solo eso, aprovechar esa cantidad de datos para obtener gráficos personalizados con los que estudiar y analizar los informes registrados.

Con esta ‘predicción’, no se pretende que una *inteligencia artificial* tome la decisión por un médico, si no que le sirva de apoyo a la hora de obtener una conclusión.

“Las máquinas me sorprenden con mucha frecuencia” (Alan Turing)

1.2 OBJETIVOS

“Aprende del pasado, vive el presente y trabaja para el futuro”.

La finalidad de este trabajo es la de realizar una herramienta verdaderamente útil y funcional, que gestione y aproveche miles de datos almacenados de pacientes. Bajo esta demanda principal podríamos definir los siguientes aspectos para alcanzarla:

- Crear una aplicación web que integre las funcionalidades necesarias para el estudio de los datos, el diseño debe ser claro e intuitivo, en una sola vista y con un par de clics de ratón obtener lo que se busca.
- Gestionar una base de datos de informes médicos con los que operar. La incorporación de esta información debe ser sencilla, seleccionar los archivos Excel donde el hospital guarda los informes y de manera automática añadirlos a nuestra base de datos.
- Diseñar una red neuronal artificial, la cual, mediante el aprendizaje y entrenamiento obtendrá predicciones del riesgo de reingreso de un paciente si es dado de alta.
- Mostrar mediante gráficas los diferentes datos que se disponen, así como poder hacer selecciones para su estudio. Acotar edades, periodos de tiempo, patologías, motivos de ingreso y grupo de diagnóstico.
- Guardar en formato PDF las diferentes gráficas obtenidas.
- Visualizar informes bajo demanda.
- Gestionar una base de datos de usuarios que puedan acceder a la aplicación. Añadir, modificar y borrar personal autorizado a utilizar la aplicación.
- Acceso mediante contraseña cifrada a la plataforma.

1.3 ESTADO DEL ARTE

La inteligencia artificial y, más concretamente, las redes neuronales artificiales están teniendo en los últimos años un gran desarrollo e impacto en diversas áreas del conocimiento, incluida la medicina.

Existen numerosos estudios en medicina donde se utiliza esta técnica para analizar todo tipo de datos en relación a este campo.

En cuanto a herramientas concretas, existe SAVANA¹, una plataforma de inteligencia artificial. La Red de Investigación de SAVANA constituye un sistema formado por más de 150 hospitales en Europa y América, concretamente en España hay más de 40 hospitales que hacen uso de ella.

¿En qué consiste y cómo funciona? Recoge la información de las historias clínicas que cada día introducen los médicos. Con el material recogido es capaz de generar una base de datos armónica entre los centros, incluso en diferentes idiomas y diferentes países. Con esta base de datos, los médicos pueden llevar a cabo investigaciones tanto retrospectivas como prospectivas.

No conozco si específicamente tiene la posibilidad de determinar el riesgo de retorno de un paciente, aunque todo apunta a que sí. No obstante, la finalidad de esta aplicación está más orientada a que sea sencilla e inmediata, es decir, poder utilizarla para hacer predicciones en el mismo momento.

1.4 PLAN DE TRABAJO

Una vez tomada la decisión de afrontar este proyecto, se estableció un plan de trabajo por fases, cada una de ellas llevaba establecido un objetivo, la consecución del objetivo marcaba la finalización de la fase y el comienzo de la siguiente.

A continuación, se muestra la tabla de fases, objetivos y tiempos de ejecución.

FASE	ACCIONES	Horas
1	Establecer objetivos y funcionalidades del proyecto	20
2	Estudio de los datos e historiales clínicos.	15
3	Estudio de redes neuronales, funcionamiento y tecnologías.	35
4	Implementación de la red neuronal Python.	45
5	Tratamiento de información y base de datos.	40
6	Diseño de la estructura de la aplicación web.	70
7	Implementación de la importación y listado de datos.	50
8	Implementación de gestión de usuarios.	55
9	Implementación de gráficas de estudio.	50
10	Implementación de predicciones.	65
11	Implementación de generación de archivos PDF.	40
12	Memoria del proyecto.	65
		550

2020

octubre - 2020							noviembre - 2020							diciembre - 2020						
Lun	Mar	Mié	Jue	Vier	Sá	Dom	Lun	Mar	Mié	Jue	Vier	Sá	Dom	Lun	Mar	Mié	Jue	Vier	Sá	Dom
			1	2	3	4							1		1	2	3	4	5	6
5	6	7	8	9	10	11	2	3	4	5	6	7	8	7	8	9	10	11	12	13
12	13	14	15	16	17	18	9	10	11	12	13	14	15	14	15	16	17	18	19	20
19	20	21	22	23	24	25	16	17	18	19	20	21	22	21	22	23	24	25	26	27
26	27	28	29	30	31		23	24	25	26	27	28	29	28	29	30	31			
							30													

2021

enero - 2021							febrero - 2021							marzo - 2021						
Lun	Mar	Mié	Jue	Vier	Sá	Dom	Lun	Mar	Mié	Jue	Vier	Sá	Dom	Lun	Mar	Mié	Jue	Vier	Sá	Dom
				1	2	3	1	2	3	4	5	6	7	1	2	3	4	5	6	7
4	5	6	7	8	9	10	8	9	10	11	12	13	14	8	9	10	11	12	13	14
11	12	13	14	15	16	17	15	16	17	18	19	20	21	15	16	17	18	19	20	21
18	19	20	21	22	23	24	22	23	24	25	26	27	28	22	23	24	25	26	27	28
25	26	27	28	29	30	31								29	30	31				
abril - 2021							mayo - 2021							junio - 2021						
Lun	Mar	Mié	Jue	Vier	Sá	Dom	Lun	Mar	Mié	Jue	Vier	Sá	Dom	Lun	Mar	Mié	Jue	Vier	Sá	Dom
			1	2	3	4						1	2		1	2	3	4	5	6
5	6	7	8	9	10	11	3	4	5	6	7	8	9	7	8	9	10	11	12	13
12	13	14	15	16	17	18	10	11	12	13	14	15	16	14	15	16	17	18	19	20
19	20	21	22	23	24	25	17	18	19	20	21	22	23	21	22	23	24	25	26	27
26	27	28	29	30			24	25	26	27	28	29	30	28	29	30				
							31													

2 INTRODUCTION

In the present that we are living, in the midst of a global pandemic, we realize that the management of health resources is of vital importance.

Unfortunately, we have had to watch on television and even suffer ourselves the collapse of many hospitals.

Given this overflow, we could raise the debate of who have priority to be cared for when there are not enough resources for everyone, first younger and then, older? It would be an endless debate, everyone would have their opinion, some from the social point of view, others from the religious, moral, political and many others. All valid, why not.

Who should have the power to choose in these situations?

3 ESPECIFICACIÓN DE REQUISITOS.

En este capítulo se describen los actores y módulos que conforman la aplicación a desarrollar.

3.1 ACTORES

Se establecen los siguientes actores:

- **Usuario:** es el personal sanitario que atiende a los pacientes y realiza el diagnóstico.
- **Administrador:** persona designada para gestionar los usuarios y los datos a importar.

3.2 MÓDULOS

Para agrupar los casos de uso establecemos dos módulos con los mismos nombres, **administrador** y **usuario**, los cuales agruparán las funcionalidades de cada actor.

3.2.1 MÓDULO ADMINISTRADOR

En la figura 3.2.1 se muestra el diagrama con los casos de uso referidos al módulo **Administrador**, engloba las acciones a realizar por dicho actor.

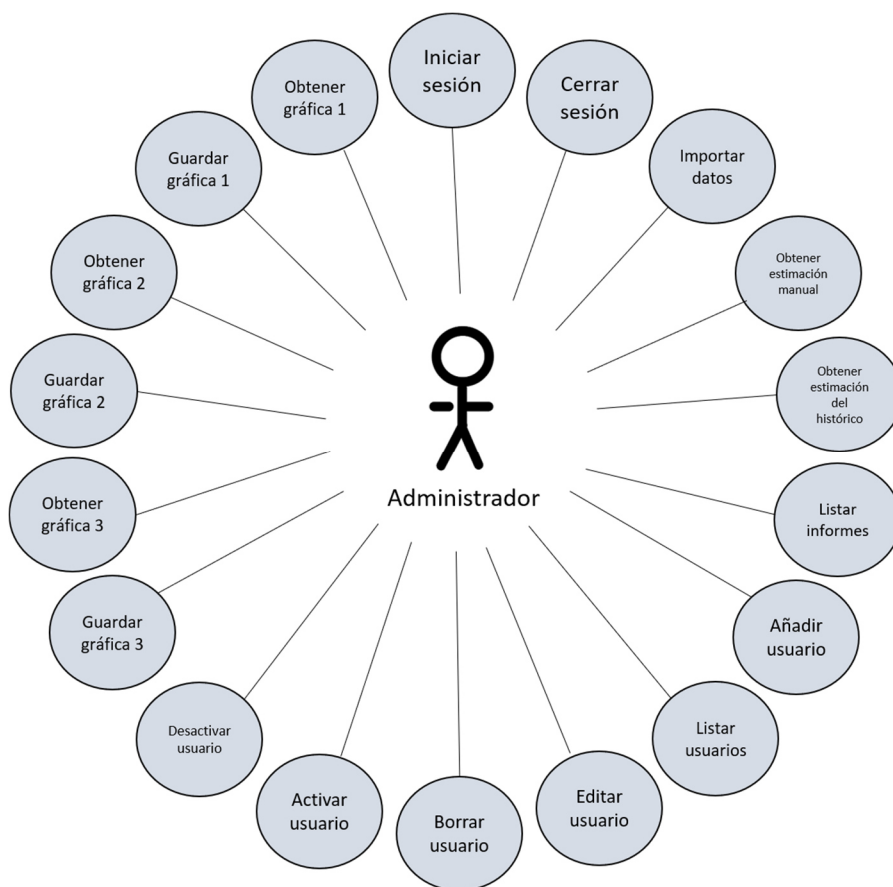


Figura 3.2.1 Diagrama de casos de uso del módulo Administrador

3.2.2 MÓDULO USUARIO

En la *figura 3.2.2* se muestra el diagrama con los casos de uso referidos al módulo **Administrador**, engloba las acciones a realizar por dicho actor.

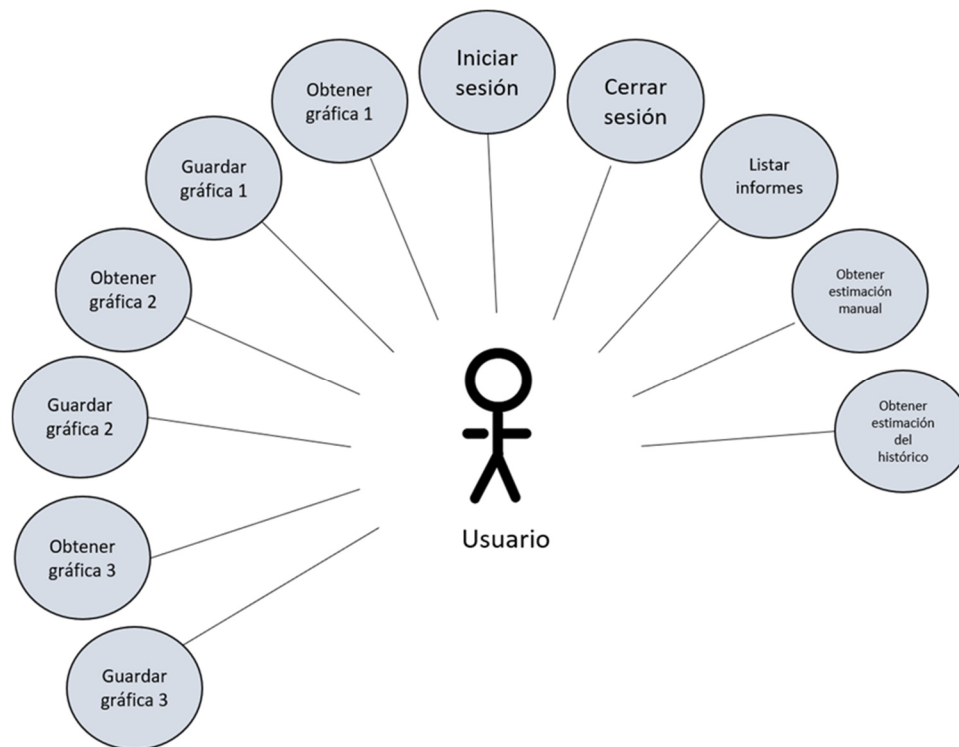


Figura 3.2.2 Diagrama de casos de uso del módulo Usuario

3.3 CASOS DE USO

En las tablas siguientes se describen los casos de uso de los dos módulos, al ser la mayoría común para ambos actores, se especifica en el apartado “Actor” de cada tabla a quién corresponde.

C01	Iniciar sesión
Actor	Administrador y Usuario
Descripción	El usuario introduce un nombre y una clave para poder acceder a la aplicación.
Precondición	Debe estar registrado previamente en el sistema y estar en estado activo.
Secuencia normal	1. Se rellena los campos “USUARIO” y “CONTRASEÑA” 2. Se pulsa el botón “VERIFICAR” 3. El sistema informa si la verificación es correcta y se redirecciona a la página inicial.
Postcondición	La persona queda habilitada para acceder a las herramientas correspondientes.
Excepciones	Paso 3: si los datos introducidos no son correctos o la persona no está en activo no accederá a la página de inicio.
Comentarios	En caso de no poder acceder se especificará mediante un mensaje en pantalla el motivo.

C02	Cerrar sesión
Actor	Administrador y Usuario
Descripción	El usuario cierra la sesión en la aplicación
Precondición	Debe estar registrado previamente en el sistema.
Secuencia normal	1. Se pulsa botón "CERRAR SESIÓN" 2. Se vuelve a la pantalla de registro de inicio
Postcondición	La aplicación pasa a la pantalla inicial de inicio de sesión
Excepciones	
Comentarios	

C03	Importar datos
Actor	Administrador
Descripción	Importa los datos de pacientes de un archivo 'Excel'.
Precondición	Se deberá tener el archivo físico en el formato correcto "xls".
Secuencia normal	1. Se selecciona la pestaña "IMPORTAR". 2. Pinchamos con el ratón en la celda "Seleccionar archivo". 3. Se abre el explorador de archivos. 4. Seleccionamos el archivo deseado. 5. Pulsar el botón "Cargar". 6. Ejecuta el proceso y se muestra un mensaje con el número de informes nuevos incorporados.
Postcondición	Nuestro sistema dispone de la información.
Excepciones	Paso 4: solo selecciona archivos Excel. Paso 5: si los expedientes a insertar ya se encuentran en la base de datos no se añaden.
Comentarios	


C04	Obtener estimación manual
Actor	Administrador y Usuario
Descripción	Evalúa un informe hipotético creado manualmente.
Precondición	
Secuencia normal	1. Se selecciona la pestaña "PREDICCIONES ur72 MANUAL" 2. Se rellenan manualmente los campos Edad, Sexo, Triage, Motivo de Ingreso, Patología y Grupo diagnóstico. 3. Se pulsa el botón "Analizar". 4. Si existe algún campo sin completar o incorrecto el sistema informa. 5. La herramienta genera con la información obtenida, el porcentaje de retorno del paciente a urgencias en las siguientes 72 horas.
Postcondición	Muestra el dato obtenido en el campo "RIESGO".
Excepciones	Paso 1: es posible no especificar alguno de los campos.
Comentarios	


C05	Obtener estimación del histórico
Actor	Administrador y Usuario
Descripción	Evalúa un informe del histórico de la base de datos.
Precondición	El informe debe estar almacenado en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. Se selecciona la pestaña “PREDICCIONES ur72 HISTÓRICO” 2. Se introduce el ID del informe a evaluar y se pulsa el botón “Buscar” para traer los datos. 3. Se pulsa el botón “Analizar”. 4. La herramienta genera con la información obtenida, el porcentaje de retorno del paciente a urgencias en las siguientes 72 horas.
Postcondición	Muestra el dato obtenido en el campo “RIESGO”.
Excepciones	Paso 1: si el ID introducido no pertenece a un registro almacenado, se mostrará un mensaje en pantalla.
Comentarios	


C06	Listar informes
Actor	Administrador y Usuario
Descripción	Muestra el histórico de informes.
Precondición	Los informes deben estar almacenados en la base de datos.
Secuencia normal	1. Se selecciona la pestaña “HISTORIALES”
Postcondición	Muestra el listado de informes.
Excepciones	No se mostrarán informes si la base de datos está vacía.
Comentarios	Se puede ordenar el listado por cualquiera de los campos mostrados.


C07	Listar Usuarios
Actor	Administrador
Descripción	Muestra un listado con la información de los usuarios de la aplicación.
Precondición	Los datos de los usuarios deben estar almacenados en la base de datos.
Secuencia normal	1. Se selecciona la pestaña “USUARIOS”
Postcondición	Muestra el listado de usuarios.
Excepciones	No se mostrarán usuarios si la base de datos está vacía.
Comentarios	Se puede ordenar el listado por cualquiera de los campos mostrados.

C08	Añadir Usuario
Actor	Administrador
Descripción	Añade un nuevo Usuario a la base de datos.
Precondición	El usuario no debe estar ya registrado en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. Se selecciona la pestaña "USUARIOS". 2. Se pulsa el botón "Añadir nuevo". 3. Se rellenan los campos requeridos: <ul style="list-style-type: none"> • <i>Nombre</i> • <i>Primer Apellido</i> • <i>Segundo Apellido</i> • <i>Contraseña</i> • <i>Repita la Contraseña</i> • <i>Ocupación</i> • <i>Tipo</i> • <i>Estado</i> 4. Se pulsa el botón "Registrar". 5. Se mostrará un mensaje en pantalla si se ha realizado correctamente. 6. Se pulsa el botón "OK"
Postcondición	El nuevo usuario se mostrará en el listado.
Excepciones	Paso 5: aparecerá un mensaje de error si: <ul style="list-style-type: none"> • Falta algún campo por completar. • Los campos "Contraseña" y "Repita la Contraseña" no son iguales. • El usuario ya existe en la base de datos.
Comentarios	

C09	Editar Usuario
Actor	Administrador
Descripción	Edita la información de un Usuario del listado.
Precondición	El usuario debe estar ya registrado en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. Se selecciona la pestaña "USUARIOS". 2. Se pulsa el icono editar . 3. Se rellenan alguno, todos o ninguno de los campos que se desean cambiar. 4. Se pulsa el botón "Modificar". 5. Se mostrará un mensaje en pantalla si se ha realizado correctamente. 6. Se pulsa el botón "OK"
Postcondición	Se mostrarán los nuevos datos del Usuario en el listado.
Excepciones	<ul style="list-style-type: none"> • Paso 5: aparecerá un mensaje de error si los campos "Contraseña" y "Repita la Contraseña" no son iguales. • Paso 5: aparecerá un mensaje de advertencia si no se ha modificado ningún campo.
Comentarios	

C10	Activar Usuario
Actor	Administrador
Descripción	El usuario ya está registrado en la base de datos y es autorizado a utilizar la aplicación.
Precondición	El usuario debe estar ya registrado en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. Se selecciona la pestaña "USUARIOS". 2. Se pulsa el icono activar . 3. Aparece un mensaje de confirmación. 4. Se acepta o cancela la activación. 5. Se mostrará un mensaje en pantalla si se ha realizado correctamente. 6. Se pulsa el botón "OK".
Postcondición	Se mostrará el nuevo estado del Usuario en el listado.
Excepciones	
Comentarios	

C11	Desactivar Usuario
Actor	Administrador
Descripción	Se deshabilita al usuario a utilizar la aplicación.
Precondición	El usuario debe estar ya registrado en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. Se selecciona la pestaña "USUARIOS". 2. Se pulsa el icono desactivar . 3. Aparece un mensaje de confirmación. 4. Se acepta o cancela la desactivación. 5. Se mostrará un mensaje en pantalla si se ha realizado correctamente. 6. Se pulsa el botón "OK".
Postcondición	Se mostrará el nuevo estado del Usuario en el listado.
Excepciones	
Comentarios	

C12	Borrar Usuario
Actor	Administrador
Descripción	Se eliminan los datos de un usuario.
Precondición	El usuario debe estar registrado en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. Se selecciona la pestaña "USUARIOS". 2. Se pulsa el icono borrar . 3. Aparece un mensaje de confirmación con los datos de Usuario. 4. Se acepta o cancela el borrado. 5. Se mostrará un mensaje en pantalla si se ha realizado correctamente. 6. Se pulsa el botón "OK".
Postcondición	Se mostrará el listado sin el Usuario borrado.
Excepciones	
Comentarios	

C13	Obtener gráfica 1
Actor	Administrador y Usuario
Descripción	Muestra una gráfica del tipo “barras”, por edades y sexo de los informes registrados.
Precondición	Debe haber informes en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. Se selecciona la pestaña “GRÁFICAS 1”. 2. Modificar, si se desea, alguno de los siguientes parámetros: <ul style="list-style-type: none"> • Edad • Fecha • Motivo de ingreso • Patología • Grupo diagnóstico 3. Pulsar el botón “Mostrar”.
Postcondición	Se carga en la página la gráfica determinada con los ajustes establecidos.
Excepciones	Paso 2: no es obligatorio modificar los valores.
Comentarios	

C14	Guardar gráfica 1
Actor	Administrador y Usuario
Descripción	Guarda en un archivo PDF los valores y la gráfica 1.
Precondición	Debe haber informes en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. Se selecciona la pestaña “GRÁFICAS 1”. 2. Pulsar el botón “GUARDAR”.
Postcondición	Se genera el archivo PDF.
Excepciones	
Comentarios	En caso de no haber una gráfica mostrada, se genera el archivo solo con los datos.

C15	Obtener gráfica 2
Actor	Administrador y Usuario
Descripción	Muestra dos gráficas del tipo “pastel”. Una representa las patologías y la otra los motivos de ingreso.
Precondición	Debe haber informes en la base de datos.
Secuencia normal	<ol style="list-style-type: none"> 1. Se selecciona la pestaña “GRÁFICAS 2”. 2. Modificar, si se desea, alguno de los siguientes parámetros: <ul style="list-style-type: none"> • Edad • Fecha 3. Pulsar el botón “Mostrar”.
Postcondición	Se carga en la página las gráficas con los ajustes establecidos.
Excepciones	Paso 2: no es obligatorio modificar los valores.
Comentarios	

C16	Guardar gráfica 2
Actor	Administrador y Usuario
Descripción	Guarda en un archivo PDF los valores y la gráfica 2.
Precondición	Debe haber informes en la base de datos.
Secuencia normal	1. Se selecciona la pestaña "GRÁFICAS 2". 2. Pulsar el botón "GUARDAR".
Postcondición	Se genera el archivo PDF.
Excepciones	
Comentarios	En caso de no haber una gráfica mostrada, se genera el archivo solo con los datos.

C17	Obtener gráfica 3
Actor	Administrador y Usuario
Descripción	Muestra una gráfica del tipo "lineal", de los ingresos y retornos mensuales.
Precondición	Debe haber informes en la base de datos.
Secuencia normal	1. Se selecciona la pestaña "GRÁFICAS 3". 2. Modificar, si se desea, alguno de los siguientes parámetros: <ul style="list-style-type: none"> • Edad • Fecha • Motivo de ingreso • Patología • Grupo diagnóstico 3. Pulsar el botón "Mostrar".
Postcondición	Se carga en la página la gráfica determinada con los ajustes establecidos.
Excepciones	Paso 2: no es obligatorio modificar los valores.
Comentarios	

C18	Guardar gráfica 3
Actor	Administrador y Usuario
Descripción	Guarda en un archivo PDF los valores y la gráfica 3.
Precondición	Debe haber informes en la base de datos.
Secuencia normal	1. Se selecciona la pestaña "GRÁFICAS 3". 2. Pulsar el botón "GUARDAR".
Postcondición	Se genera el archivo PDF.
Excepciones	
Comentarios	En caso de no haber una gráfica mostrada, se genera el archivo solo con los datos.

4 TECNOLOGÍAS EMPLEADAS

En este apartado se describen las diferentes tecnologías empleadas en cada una de las partes del proyecto.

Para el control de versiones, al ser realizado por una sola persona, no se ha requerido de ningún programa especial. Simplemente, después de cada jornada de trabajo se guardaban dos copias de la carpeta del proyecto. Una en el disco local y otra en **Google Drive**². Estas copias eran nombradas con la fecha del día correspondiente.

Para la edición del código se ha utilizado la herramienta “**Visual Studio Code**”³ desarrollado por Microsoft. Las razones para su elección son varias:

- Incluye soporte para la depuración.
- Resaltado de sintaxis.
- Finalización inteligente de código.
- Es personalizable.
- Es gratuito y de código abierto.

Como entorno de desarrollo se emplea **XAMPP**⁴, una distribución de **Apache**⁵ que contiene **MariaDB**⁶ y **PHP**⁷. Los motivos son, que es gratuita y fácil de instalar y usar.

4.1 APLICACIÓN WEB

- Para el desarrollo de la aplicación web, se han utilizado diversos lenguajes como **HTML5**⁸, **CSS3**⁹, **JavaScript**¹⁰ y **PHP**.
- Se ha utilizado JavaScript para el control de errores y la generación de gráficas.
- Mediante PHP se ha conectado la BD con la aplicación web y realizado diferentes formularios, entre otras cosas.
- Para el framework se ha utilizado **AdminLTE**¹¹, un panel de administración para **Bootstrap**¹². Es una solución de código abierto basada en un diseño modular que permite una construcción y personalización sencillas.
- Se ha buscado la sencillez y la facilidad a la hora de realizar el diseño, es otra razón para escoger las metodologías citadas, ya que cuentan con un gran soporte de la comunidad informática.

4.2 RED NEURONAL

Hoy en día está muy extendido el uso de redes neuronales para sistemas de predicción, por ello es fácil encontrar gran cantidad de documentación y ejemplos de las mismas.

En muchos de los casos estudiados se empleaba código **Python**¹³ para su implementación, por ello me decante por este lenguaje. Además de contar con librerías explícitas a este fin, es un lenguaje de estilo flexible y sencillo, sin contar con experiencia previa en él, como era mi caso, en poco tiempo se adquiere la destreza para utilizarlo.

4.3 BASE DE DATOS

Para administrar los datos se utiliza el sistema de gestión **MariaDB**, el cual deriva de **MySQL**¹⁴, viene integrado en XAMPP, es uno de los más populares y utilizados en los entornos de desarrollo web.

El control de la gestión lo realizamos con **phpMyAdmin**¹⁵, una herramienta escrita en PHP con la intención de manejar la administración de MariaDB a través de páginas web. Por medio del navegador web integra de manera sencilla todas funcionalidades necesarias para trabajar con bases de datos.

5 ARQUITECTURA DE LA APLICACIÓN

Se ha desarrollado una arquitectura *cliente-servidor* por ser un modelo flexible y adaptable al servicio que se quiere implementar. Este nos permite aumentar el rendimiento, así como también, envolver variadas plataformas, bases de datos, redes y sistemas operativos que pueden ser de diferentes distribuidores con arquitecturas totalmente diferentes y funcionando todos al mismo tiempo.

Además, se puede considerar un sistema ventajoso en cuanto a seguridad, ya que el servidor controla el acceso a sus datos por lo que se necesita que el servidor nos autorice para poder acceder a él.

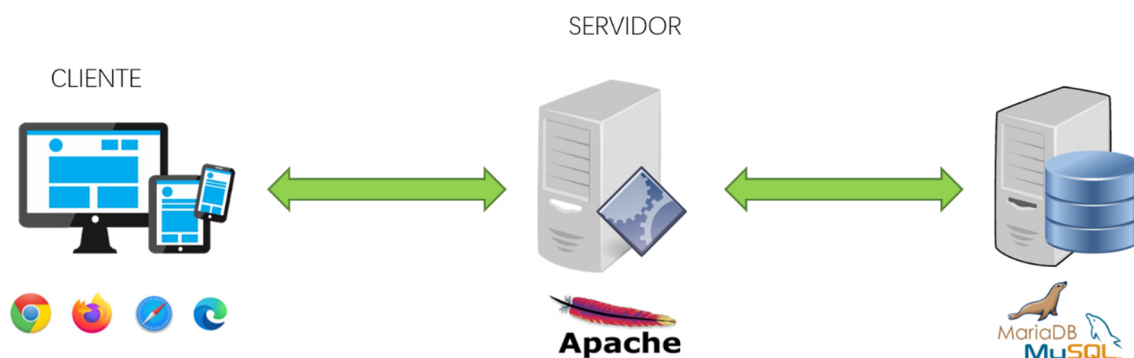
El proyecto no cuenta con un servidor en la nube como tal, para realizar las pruebas se utilizó **XAMPP**, una herramienta que permite probar desarrollo web basado en PHP en tu propio ordenador sin necesidad de tener acceso a internet.

Cuenta con la ventaja de que no es necesario saber sobre las configuraciones de servidores, ya que **XAMPP** te provee de una configuración totalmente funcional desde el momento que lo instalas. Básicamente lo extraes y listo.

El paquete del software contiene los mismos componentes que se utilizan en cualquier servidor web, de forma que permite testar proyectos localmente y transferirlos cómodamente a sistemas reales.

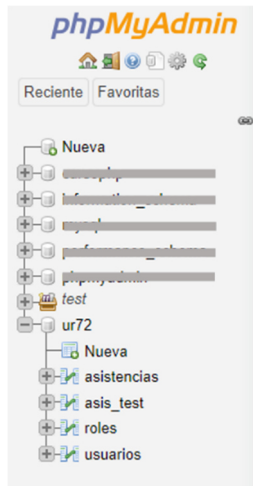
Dentro de **XAMPP**, contamos con la herramienta **phpMyAdmin**, con la que podemos administrar la base de datos comentada en el apartado anterior.

La parte del cliente la representa el **usuario**, que a través de un ordenador, smartphone o Tablet accede a la aplicación.



6 MODELO DE DATOS

En esta sección se describe la estructura y tipo de datos de la base de datos y la forma en que se relacionan. Para implementar el proyecto se ha definido una base de datos denominada “ur72”, la cual incluye las tablas mostradas en la figura 5.0.



- **asistencias**: recoge los historiales de pacientes generados por el hospital.
- **usuarios**: con los datos personales de registro de los usuarios.
- **roles**: define el tipo de usuario.
- **asis_test**: tabla auxiliar para realizar las predicciones.

Figura 5.0

6.1 Tabla “asistencias”

Esta tabla almacena los historiales de pacientes, por tanto, los campos vienen ya establecidos por el formato que tiene el hospital para recoger la información de un paciente tratado en el servicio de urgencias.

Los datos son registrados por el personal sanitario en una hoja EXCEL y se importan a la siguiente tabla mediante una funcionalidad de la aplicación.

Son un total de 62 campos, los cuales no tienen que rellenarse en su totalidad, dependiendo del tipo de incidencia se completan o no.

A continuación, se muestra una tabla descriptiva con el nombre, tipo y funcionalidad de cada uno de los parámetros.

#	Nombre	Tipo	Funcionalidad
1	NHC	int	Número de historia del paciente
2	ID_ASISTENCIA	int	Identificador único para cada asistencia
3	EDAD	int	Edad del paciente
4	SEXO	text	Sexo del paciente
5	LOCALIDAD	int	Localidad
6	ESTADO_ACTUAL_EXITUS	text	Indica si el paciente ha fallecido
7	MEDICO_FAMILIA	int	Médico de familia del paciente
8	PATOLOGIA	text	Patología
9	FECHA_INGRESO	datetime	Fecha de ingreso del paciente

10	MOTIVO_INGRESO	text	Causa del ingreso
11	FECHA_TRIAJE	datetime	Fecha y hora registrada en sala de triaje
12	FECHA_EMERGENCIAS	datetime	Fecha y hora de entrada del paciente a sala de emergencias
13	FECHA_PRIMERA_CONSULTA	datetime	Fecha y hora de la primera atención por parte del médico
14	FECHA_CONSULTA	datetime	Fecha de la consulta
15	TIEMPO_TRIAJE_MIN	int	Espera de triaje
16	TIEMPO_EMERGENCIA_MIN	int	Tiempo que tarda en pasar un paciente al estado de emergencia
17	TIEMPO_CONSULTA_MIN	int	Espera de primera consulta
18	TIEMPO_OBS_MIN	int	Espera a ingreso en observación
19	TIEMPO_TCO_MIN	int	Espera a ingreso en sala de tratamientos cortos
20	TIEMPO_ALTA_MEDICA_MIN	int	Tiempo total hasta el alta médica
21	TIEMPO_ALTA_MIN	int	Tiempo hasta el alta administrativa
22	PRIMER_MEDICO_ASIGNADO	int	Primer médico asignado
23	PRIMER_MEDICO_DE_CONSULTA	int	Primer médico con el que tiene una consulta
24	MEDICO_ALTA	int	Médico que le da el alta al paciente
25	MOTIVO_CONSULTA	text	Motivo de la consulta
26	NIVEL_TRIAJE	int	Nivel de triaje
27	TENS_ARTERIAL_MAX	int	Tensión arterial máxima
28	TENS_ARTERIAL_MIN	int	Tensión arterial mínima
29	FRECUENCIA_CARDIACA	int	Frecuencia cardiaca
30	SATURACION_O2	int	Saturación de oxígeno recogida en triaje
31	TEMPERATURA	int	Temperatura recogida en triaje
32	GLUCEMIA	int	Nivel de glucosa en sangre recogida en triaje
33	EVA	int	Escala de dolor recogida en triaje
34	GLASGOW	int	Escala que valora el nivel de conciencia, recogida en triaje
35	COMENTARIO	text	Texto libre del enfermero de triaje
36	MOTIVO_ALTA	text	Motivo del alta
37	PROCESOS	text	Proceso asignado en base a catálogo asistencial
38	JUICIO_CLINICO	text	Juicio clínico de campo libre
39	ADECUACION_CONSULTA	text	Adecuación de consulta
40	PROCEDIMIENTOS	text	Procedimientos realizados en base al catálogo asistencial
41	REVISION_DE_URG	text	Derivación a consulta especializada desde urgencias
42	CONCILIACION_MEDICAMENTOSA	int	Se valora en el alta del tratamiento basal del paciente
43	FECHA_ALTA	datetime	Fecha y hora del alta
44	FECHA_ALTA_MEDICA	datetime	Fecha y hora del alta médica
45	DESTINO	text	Descripción detallada del motivo del alta
46	DIAGNOSTICO_PPAL	text	Diagnóstico principal en base a CIE-9
47	GRUPO_DIAGNOSTICO	text	Grupo de diagnóstico en base a CIE-9
48	RECONSULTAS_72	int	Número de re-consultas después de 72 horas tras el alta
49	RECONSULTAS_ULT_ANIO	int	Número de visitas a urgencias en el último año
50	RETRIAJES	int	El sistema vuelve a realizar el triaje si el paciente cambia su situación clínica
51	ICCAE_SN	text	Tiene informe de continuidad de cuidados
52	PRIMERA_FECHA_SOL_LAB	datetime	Primera fecha de solicitud al laboratorio
53	PRIMERA_FECHA_SOL_RAD	datetime	Primera fecha de solicitud a radiología
54	MEDIO_TRANSPORTE_INGRESO	text	Medio de transporte del ingreso
55	TRANSPORTE_SN	text	Acude al centro en transporte
56	INCIDENCIAS	text	Incidencias
57	TRANSFUSIONES	text	Recibe transfusión

58	CAMA_OBS	int	Cama concreta del área de observación
59	CAMA_TCO	int	Cama concreta del área de tratamientos cortos
60	INTERVENCION QUIRURGICA	text	Intervención quirúrgica
61	INTERCONSULTA	text	Interconsulta
62	ENTIDAD	text	Entidad que asegura al paciente

6.2 Tabla “usuarios”

Tabla que administra los datos de los usuarios registrados para utilizar la aplicación, todos los campos deben ser completados en el formulario de inscripción.

Como características principales se destacan las siguientes:

- **usu_id**: identifica con un número autogenerado a la persona, es único.
- **usu_nombre**: se establece el valor de este campo como único, por tanto, no pueden registrarse dos usuarios con el mismo nombre. Se define así porque se toma este dato para acceder a la aplicación y no sería posible determinar que usuario quiere entrar si hay varios con el mismo nombre. Se sugiere que si hay personas con idéntico nombre se le añada un número (Ana, Ana1, Ana2 ...).
- **rol_id**: referencia el valor “rol_id” de la tabla “rol”.

A continuación, se muestra una tabla descriptiva con el nombre, tipo y funcionalidad de cada uno de los parámetros.

#	Nombre	Tipo	Funcionalidad
1	usu_id 🔑	int	Id único de usuario
2	usu_nombre 🔑	text	Nombre de usuario
3	usu_apellido1	text	Primer apellido del usuario
4	usu_apellido2	text	Segundo apellido del usuario
5	usu_pwd	varchar	Contraseña del usuario
6	usu_empleo	varchar	Labor del usuario en el centro
7	rol_id 🔑	int	Tipo de usuario
8	usu_estatus	enum	‘ACTIVO’ o ‘INACTIVO’

6.3 Tabla “rol”

Se crea esta tabla con el fin de establecer ciertos privilegios en función del rol del usuario que accede a la aplicación.

- **rol_id:** es un valor numérico que referencia un campo del mismo nombre en la tabla *usuarios*. Es único y auto incrementable.
- **rol_nombre:** se han definido dos ‘ADMIN’ y ‘USER’ que corresponden a los actores de la especificación de requisitos del apartado 2.1.

A continuación, se muestra una tabla descriptiva con el nombre, tipo y funcionalidad de cada uno de los parámetros.

#	Nombre	Tipo	Funcionalidad
1	rol_id 🔑	int	Id único del rol
2	rol_nombre	varchar	Nombre del rol

6.4 Tabla “asis_test”

A la hora de generar una predicción sobre un expediente, ya sea del histórico o uno generado manualmente, solo se tienen en cuenta 10 de los 62 parámetros que tiene un informe original. Tampoco se utilizan el 100% de las entradas, es por estos motivos que se crea esta tabla.

Contiene solo los campos y entradas que se utilizan en la red neuronal que realiza el estudio de los datos. De esta manera agilizamos el proceso y preservamos la tabla principal de interacciones.

A continuación, se muestra una tabla descriptiva con el nombre, tipo y funcionalidad de cada uno de los parámetros.

#	Nombre	Tipo	Funcionalidad
1	NHC	int	Número de historia del paciente
2	ID_ASISTENCIA	int	Identificador único para cada asistencia
3	EDAD	int	Edad del paciente
4	SEXO	text	Sexo del paciente
5	PATOLOGIA	text	Patología
6	FECHA_INGRESO	datetime	Fecha de ingreso del paciente
7	MOTIVO_INGRESO	text	Causa del ingreso
8	NIVEL_TRIAJE	int	Nivel de triaje
9	GRUPO_DIAGNOSTICO	text	Grupo de diagnóstico en base a CIE-9
10	RECONSULTAS_72	int	Número de re-consultas después de 72 horas tras el alta

6.5 Relaciones entre tablas

En la figura 6.5.1 se muestran las relaciones entre las tablas del proyecto. La única que existe es **rol_id** entre “*usuarios*” y “*roles*”.

El resto son independientes, aunque la tabla “*asis_test*” se genera con datos de “*asistencias*”, se decide establecer una independencia entre ellas por motivos de seguridad y rendimiento.

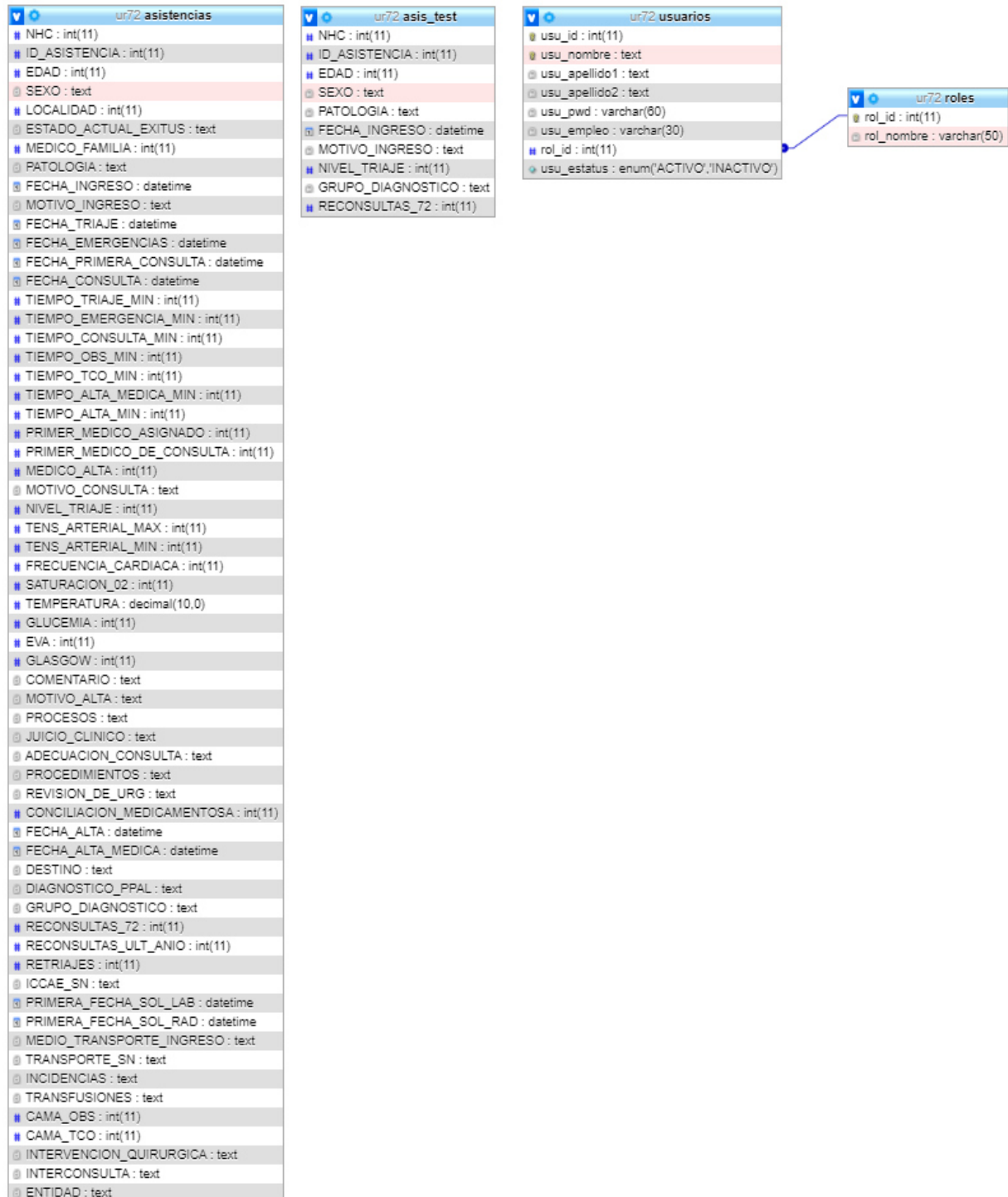


Figura 6.5.1

7 DISEÑO

Uno de los aspectos importantes en la parte del cliente para nuestra aplicación es el diseño, una buena interfaz es determinante para que el usuario pueda sacar el mayor rendimiento a nuestra herramienta.

A la hora de diseñar, en segundo lugar, después de la funcionalidad, se ha intentado buscar la sencillez en el funcionamiento y en la visualización de la información.

Poder obtener en un par de clics de ratón lo que se busca, me parecía esencial, sobre todo en una herramienta destinada a un servicio de urgencias, donde el tiempo, en la mayoría de ocasiones es un bien escaso.

Por estas razones la aplicación está formada por dos únicas páginas:

- Página inicial de registro.
- Página principal.

En cuanto a la manera de ejecutarlo se ha hecho uso de **Bootstrap**, un kit de herramientas de código abierto para desarrollos web responsive con HTML, CSS y JavaScript. Incluye todos los elementos necesarios para maquetar la página, modales, menús, cuadros, botones, formularios...

7.1 Página inicial de registro

Es lo primero que nos encontramos al acceder a la aplicación (*Figura 7.1.1*). El usuario deberá ingresar correctamente su nombre y contraseña.

Se dispone un recuadro central fácilmente identificable donde poder registrarse, el campo contraseña, oculta los caracteres ingresados para mantener la confidencialidad.

Al pulsar sobre el botón “ENTRAR”, muestra un mensaje emergente de aprobación si los datos son correctos, o un mensaje de error en el caso contrario.

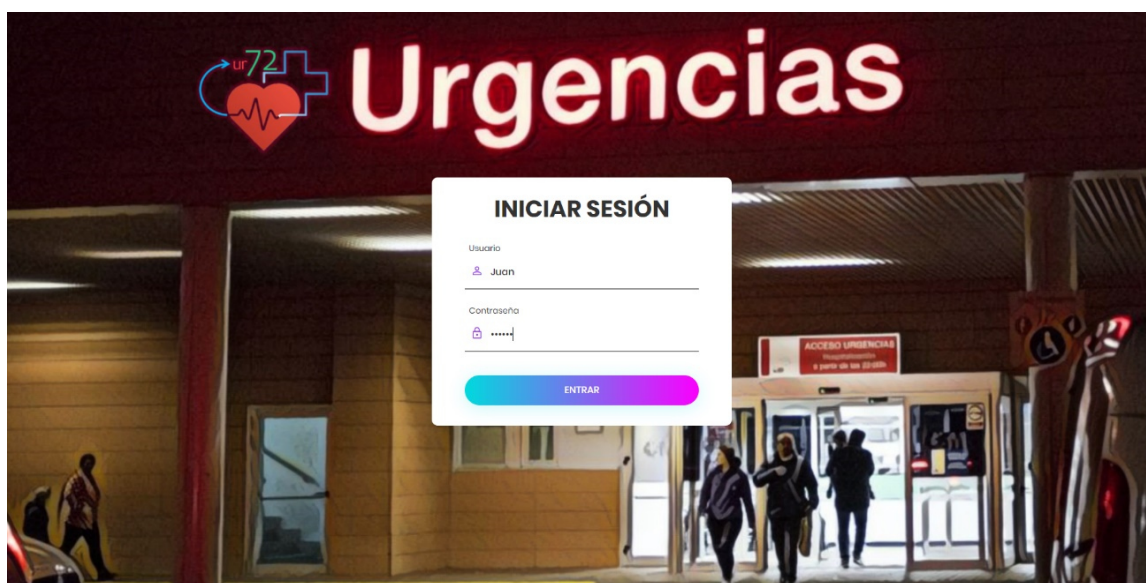


Figura 7.1.1

7.2 Página principal

Después de registrarnos correctamente nos aparecerá automáticamente esta página (figura 7.2.1), se divide en dos partes bien definidas y claramente identificables:

- Zona izquierda de la pantalla, una columna oscura destinada a un contenido meramente informativo. Muestra el logo de la aplicación, el nombre del usuario conectado y un apartado “AYUDA” donde irán aclaraciones y explicaciones.
- El resto de la pantalla de destina a las utilidades. Se han dispuesto en pestañas verticales agrupadas por colores para que sean fácilmente reconocibles. Pulsando a la derecha de la pestaña en ‘+’ se desplegará su contenido (figura 7.2.2).



Figura 7.2.1

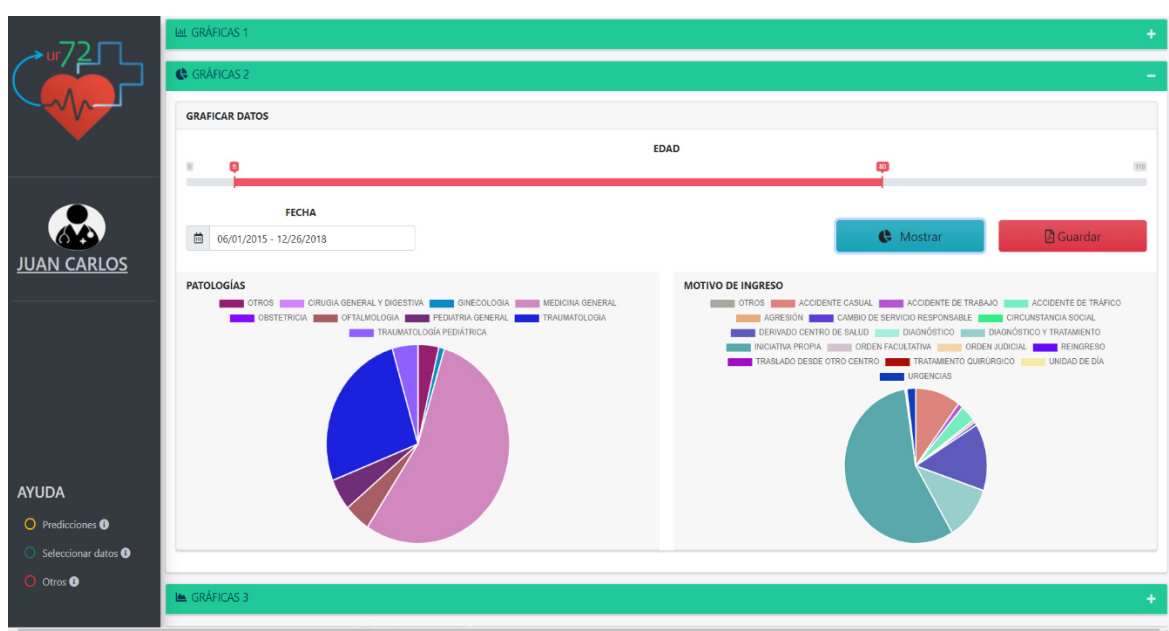


Figura 7.2.2

8 IMPLEMENTACIÓN

En esta sección se mostrará la implementación realizada en la ejecución de la aplicación, se divide en dos partes, los scripts de Python donde se realizan las predicciones de la red neuronal y una segunda parte que corresponde a la aplicación web.

8.1 IMPLEMENTACIÓN RED NEURONAL

Una de las principales razones para elegir Python a la hora de confeccionar una red neuronal fue la amplia disponibilidad de librerías que posee. Concretamente en nuestro caso utilizamos **TensorFlow**¹⁶ y **keras**¹⁷.

- **TensorFlow** es una librería de computación matemática, que ejecuta de forma rápida y eficiente gráficos de flujo.
- **Keras** es una biblioteca de redes neuronales escrita en de Python. **Keras** es una abstracción, un API High-level, para la creación de modelos de aprendizaje. Aporta una sintaxis homogénea y una interface sencilla, modular y ampliable para la creación de redes neurales.

Las redes neurales son un tipo particular de gráfico de flujo de datos. Por tanto, **TensorFlow** y **Keras** combinan perfectamente haciendo un tándem que aúna potencia, sencillez de uso y rapidez de ejecución.

La síntesis de funcionamiento del código para la red neuronal sería el siguiente:

1. Recibimos el dato que queremos evaluar.
2. Conecta con la base de datos.

```
1 # Librerías
2 import silence_tensorflow.auto
3 import tensorflow as tf
4 import pymysql
5 import sys
6 import numpy as np
7 import pandas as pd
8 from keras.models import Sequential
9 from keras.layers import Dense
10
11 # dato a analizar que se recibe
12 test_ID_generado = int(sys.argv[1])
13
14 # conexión con la base de datos
15 miConexion = pymysql.connect(host="localhost", user="root", passwd="", db="cursophp")
16
```

3. Se extraen los campos de los historiales, únicamente los que se van a utilizar en la predicción.

```
16
17 # importar datos
18 dataset = pd.read_sql_query("SELECT ID_ASISTENCIA, EDAD, SEXO, PATOLOGIA, MOTIVO_INGRE
19 datasetGen = pd.read_sql_query("SELECT ID_ASISTENCIA, EDAD, SEXO, PATOLOGIA, MOTIVO_IN
20
21 # añade el historial generado al grupo a codificar
22 dataset = dataset.append( (datasetGen), ignore_index=True)
23 dataset = dataset.sort_values('ID_ASISTENCIA')
24
```


4. Se establece el valor por defecto 'INDETERMINADO' en los sitios donde el dato esté vacío.

```
24
25 # se completan los campos vacíos
26 dataset = dataset.replace({"": 'INDETERMINADO'})
27
```

5. Se convierten los valores de tipo categórico a numérico (One Hot Encoding¹⁸).

```
27
28 # Cambiamos el dato categórico "SEXO", a numérico (one hot encoding)
29 Sexo = pd.get_dummies(dataset["SEXO"], prefix='Sexo')
30 dataset = pd.concat([dataset, Sexo], axis=1)
31 # Borramos el campo categórico
32 dataset = dataset.drop(["SEXO"], axis=1)
33
34 # Cambiamos el dato categórico "PATOLOGIA", a numérico (one hot encoding)
35 Patologia = pd.get_dummies(dataset["PATOLOGIA"], prefix='Patologia')
36 dataset = pd.concat([dataset, Patologia], axis=1)
37 # Borramos el campo categórico
38 dataset = dataset.drop(["PATOLOGIA"], axis=1)
39
40 # Cambiamos el dato categórico "MOTIVO_INGRESO", a numérico (one hot encoding)
41 M_I = pd.get_dummies(dataset["MOTIVO_INGRESO"], prefix='M_I')
42 dataset = pd.concat([dataset, M_I], axis=1)
43 # Borramos el campo categórico
44 dataset = dataset.drop(["MOTIVO_INGRESO"], axis=1)
45
46 # Cambiamos el dato categórico "GRUPO_DIAGNOSTICO", a numérico (one hot encoding)
47 G_D = pd.get_dummies(dataset["GRUPO_DIAGNOSTICO"], prefix='G_D')
48 dataset = pd.concat([dataset, G_D], axis=1)
49 # Borramos el campo categórico
50 dataset = dataset.drop(["GRUPO_DIAGNOSTICO"], axis=1)
51
```

6. Se ajustan y estructuran los datos.

```
51
52 # si ha vuelto más de una vez se considera solo 1
53 dataset.loc[dataset["RECONSULTAS_72"] > 1, "RECONSULTAS_72"] = 1
54 # recolocamos la variable objetivo en la columna final
55 Rec_72 = dataset["RECONSULTAS_72"]
56 dataset.loc[:, 'Rec_72'] = Rec_72
57 # Borramos la antigua
58 dataset = dataset.drop(["RECONSULTAS_72"], axis=1)
59
60 # guardamos el dato a evaluar codificado
61 is_P = dataset.loc[:, 'ID_ASISTENCIA'] == test_ID_generado
62 P = dataset.loc[is_P]
63 # borramos el dato a evaluar de las muestras de entrenamiento
64 dataset = dataset.drop(dataset[dataset['ID_ASISTENCIA'] == test_ID_generado].index)
65
```

7. Se establecen 3 grupos de datos:
 - I. X: variables de entrada.
 - II. Y: variables de salida.
 - III. P: variables que queremos predecir.

```
65
66 # divide en variables de entrada (X), salida (Y) y predicción (P)
67 cols = len(dataset.columns)
68 X = dataset.iloc[:, 1:(cols-1)]
69 Y = dataset.iloc[:, -1]
70 P = P.iloc[:, 1:(cols-1)]
71
```

8. Se crea la red neuronal, su arquitectura y modelo.

```
71
72 # arquitectura de la red neuronal
73 # Primero creamos un modelo vacío de tipo Sequential.
74 # Este modelo se refiere a que crearemos una serie de capas de neuronas secuenciales, "una delante de otra".
75 model = Sequential()
76
77 # capa primera cols-2 neuronas, las entradas
78 # capa oculta 1, 300 neuronas, función de activación 'relu'
79 model.add(Dense(300, input_dim=(cols-2), activation='relu'))
80 #model.add(Dense(50, input_dim=113, activation='tanh'))
81
82 # capa oculta 2, 200 neuronas, función de activación 'relu'
83 #model.add(Dense(50, activation='tanh'))
84 model.add(Dense(200, activation='relu'))
85
86 # capa de salida, 1 neurona, función de activación 'sigmoid'
87 model.add(Dense(1, activation='sigmoid'))
88 model.summary()
89
```

9. Se compila.

```
89
90 # Compila el modelo,ajustes
91 model.compile(loss='binary_crossentropy', # tipo de pérdida
92               optimizer='adam',         # optimizador de los pesos
93               metrics=['binary_accuracy'])# métrica a obtener, otra opción 'binary_accuracy'
94
95 # Conversión de tipo de datos a int
96 X = np.asarray(X).astype(np.int)
97 Y = np.asarray(Y).astype(np.int)
98 P = np.asarray(P).astype(np.int)
99
```

10. Se realiza el entrenamiento de la red.

11. Se obtiene la predicción.

```
99
100 # Entrenamiento de red
101 validation_split = 0.2 # for training
102 # entradas, salidas, entrenamientos, cada cuantos datos ajusta los pesos
103 model.fit(X, Y, epochs=50, batch_size=100)
104
105 # calcula la predicción
106 predictions = model.predict(P)
107
108 # predicción en %
109 percent = predictions[0]*100
110 # predicción número entero
111 ur72 = int(percent)
112 #print(ur72)
113
```

8.2 IMPLEMENTACIÓN APLICACIÓN WEB

A continuación, se expone la implementación de la parte web de nuestra aplicación, a la hora de organizar la lógica y el código del software, se ha optado por utilizar el patrón **MVC**¹⁹ (modelo vista controlador).

Empecemos por el apartado “vista”, el diseño está basado en una estructura de rejilla en la que, por medio de filas y columnas se colocan los contenidos. En la figura 8.2.1 se muestra la pantalla principal, está formada por columnas seriadas, en cada una de ellas se desarrollan cada una de las funcionalidades de la aplicación. De esta manera modularizamos la aplicación, pudiendo añadir o quitar secciones en función de lo requerido.

```
<body class="hold-transition sidebar-mini layout-fixed">
<script type="text/javascript" src="../js/user.js?rev=<?php echo time();?>"></script>

<!-- ***** PREDICCIONES INGRESOS MANUAL ***** -->
> <div class="col-md-12">...
</div>

<!-- ***** PREDICCIONES INGRESOS HISTÓRICO ***** -->
> <div class="col-md-12">...
</div>

<!-- ***** HISTORIAL DE INGRESOS ***** -->
> <div class="col-md-12">...
</div>

<!-- ***** LISTA DE USUARIOS ***** -->
> <div class="col-md-12">...
</div>

<!-- ***** IMPORTAR EXCEL ***** -->
> <div class="col-md-12">...
</div>

<!-- ***** GRÁFICAS 01 ***** -->
> <div class="col-md-12">...
</div>

<!-- ***** GRÁFICAS 02 ***** -->
> <div class="col-md-12">...
</div>

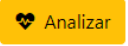
<!-- ***** GRÁFICAS 03 ***** -->
> <div class="col-md-12">...
</div>

<!-- ***** FOOTER ***** -->
> <div class="col-md-12">...
</div>
```

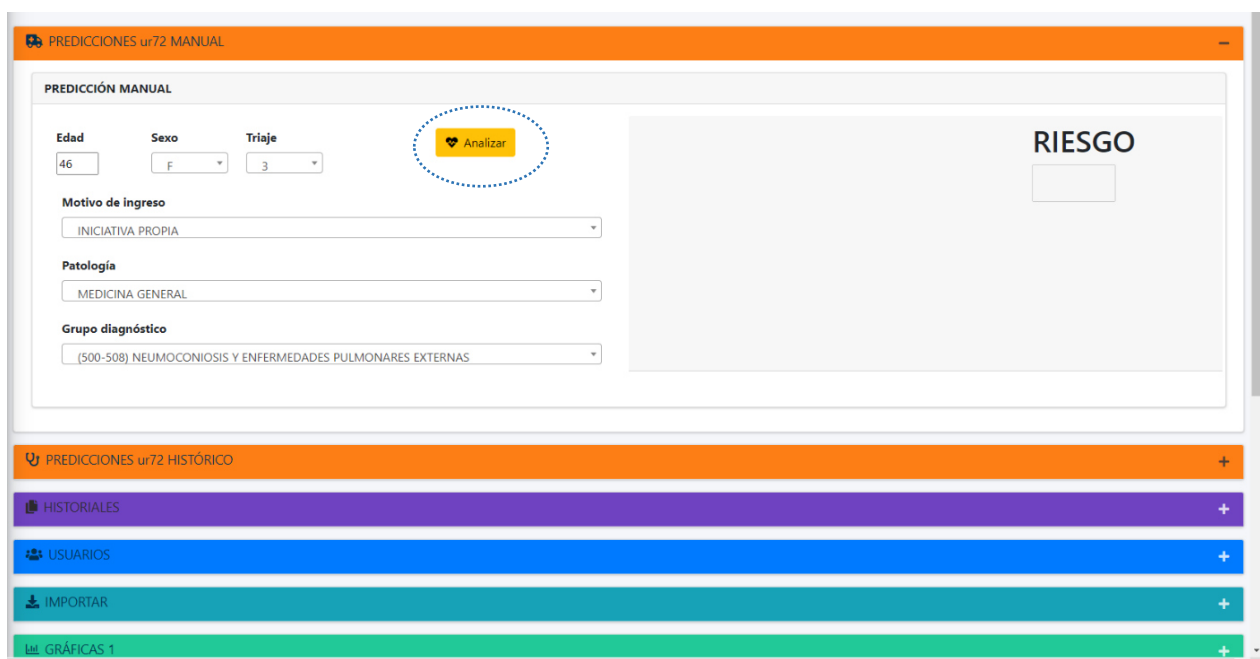
Figura 8.2.1



Dentro de cada sección distribuimos los elementos correspondientes de igual manera, utilizando columnas y filas.

En nuestro ejemplo, al pulsar el botón  , se recogen los datos de los campos por medio de la función “analizar_informe_generado ()”.

```
vista > usuario > vista_usuario.php
52 <script type="text/javascript" src="../../js/user.js?rev=<?php echo time();?>"></script>
53
54 <!-- ***** PREDICCIONES INGRESOS MANUAL ***** -->
55 <div class="col-md-12">
56 <div class="card card-orange">
57 <div class="card-header">...
58 </div>
59 <!-- /.card-header -->
60 <div class="card-body" style="display: none;">
61 <div class="col-lg-12" style="padding">
62 <div class="card">
63 <div class="card-header">
64 <b>PREDICCIÓN MANUAL</b>
65 </div>
66 <div class="card-body">
67 <div class="row">
68 <div class="col-sm-6">
69 <div class="row">
70 <div class="col-lg-12" style="padding">
71 <div class="card-body">
72 <div class="row">
73 <div class="col-lg-2"><label for="">Edad</label><br>
74 <input type="text" size="2" maxlength="2" id="paciente_edad_1">
75 </div>
76 <div class="col-lg-2"><label for="">Sexo</label><br>
77 <select class="js-example-basic-single" name="state" id="cbm_sexo_1" style="width:100%;">
78 <option value="M">M</option>
79 <option value="F">F</option>
80 </select><br>
81 </div>
82 <div class="col-lg-2"><label for="">Triaje</label><br>...
83 </div>
84 <div class="col-lg-3"><button class="btn btn-warning" style="width:100%" onclick="analizar_informe_generado()"
85 id="btn_cargar_py_1"><i class="nav-icon fas fa-heartbeat"></i>Analizar</button>
86 </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 <div class="col-lg-12"><label for="">Motivo de ingreso</label>...
92 </div>
93 <div class="col-lg-12"><label for="">Patología</label>
94 <select class="js-example-basic-single" name="state" id="cbm_pt" style="width:100%;">
95 </select><br><br>
96 </div>
97 <div class="col-lg-12"><label for="">Grupo diagnóstico</label>
98 <select class="js-example-basic-single" name="state" id="cbm_g_d" style="width:100%;">
99 </select><br><br>
100 </div>
101 </div>
102 </div>
103 </div>
104 </div>
105 </div>
106 </div>
107 </div>
108 </div>
109 </div>
110 </div>
111 </div>
112 </div>
113 </div>
114 </div>
```



Esta información, tras hacer las comprobaciones pertinentes, en este caso en particular, comprueba que el campo “EDAD” ha sido rellenado, se manda a través del método **ajax**²⁰ al controlador.

```
927 function analizar_informe_generado(){
928     var edad      = $("#paciente_edad_1").val();
929     var sexo      = $("#cbm_sexo_1").val();
930     var triaje     = $("#cbm_triaje").val();
931     var motivo    = $("#cbm_m_i").val();
932     var patologia  = $("#cbm_pt").val();
933     var grupo_diag = $("#cbm_g_d").val();
934
935     if(edad.length==0)
936         return Swal.fire("Advertencia","Rellene el campo 'EDAD',"warning");
937
938     $.ajax({
939         "url": "../controlador/user/controlador_analizar_generado.php",
940         type: 'POST',
941         data: {
942             edad: edad,
943             sexo: sexo,
944             triaje: triaje,
945             motivo: motivo,
946             patologia: patologia,
947             grupo_diag: grupo_diag
948         }
949     }).done(function(resp){
950         /* refresh gráfica */
951         if (window.myPie_man) {
952             window.myPie_man.clear();
953             window.myPie_man.destroy();
954         }
955
956         var percent = [];
957         var result = resp.split(";");
958         percent[0] = result[1];
959         percent[1] = 100 -percent[0];
960
961         $("#porcentaje2").val(percent[0]+"%");
962         var ctx = document.getElementById('myChart_man');
963         window.myPie_man = new Chart(ctx, {
964             type: 'doughnut',
965             data: {
966                 labels: ['SI','NO'],
967                 datasets: [
968                     {
969                         data: percent, backgroundColor: ['#FA361B','#D6F8CA'],
970                     }
971                 ]
972             }
973         });
974     });
975 }
```

El **controlador** sirve de enlace entre la *vista* y el **modelo**, se encarga como su nombre indica de controlar las interacciones del usuario en la *vista*, pide los datos al **modelo** y los devuelve de nuevo a la *vista* para que esta los muestre al usuario.

```

controlador > user > controlador_analizar_generado.php
1  <?php
2      require '../modelo/modelo_user.php';
3      $MU = new Modelo_Usuario();
4
5      $id = 72;
6      $edad = htmlspecialchars($_POST['edad'], ENT_QUOTES, 'UTF-8');
7      $sexo = htmlspecialchars($_POST['sexo'], ENT_QUOTES, 'UTF-8');
8      $triaje = htmlspecialchars($_POST['triaje'], ENT_QUOTES, 'UTF-8');
9      $motivo = htmlspecialchars($_POST['motivo'], ENT_QUOTES, 'UTF-8');
10     $patologia = htmlspecialchars($_POST['patologia'], ENT_QUOTES, 'UTF-8');
11     $grupo_diag = htmlspecialchars($_POST['grupo_diag'], ENT_QUOTES, 'UTF-8');
12
13     if($MU->insert_generado($id, $edad, $sexo, $triaje, $motivo, $patologia, $grupo_diag)){
14         $consulta = $MU->prediccion_generado($id);
15         echo $consulta;
16     }
17     else echo 0;
18
19     ?>

```

Las funciones del **modelo** en esta aplicación, tiene dos misiones principalmente:

- Unas conectar con la base de datos, para extraer o guardar información. (Función *ImportarDatosGrafica3_2()* de la *Figura 7.2.1*)
- Otras, como es el caso del ejemplo, ejecutar el script de Python que contiene la red neuronal y recoger el resultado. (Función *prediccion_generado()* de la *Figura 7.2.1*)

```

modelo > modelo_user.php
360
361     function ImportarDatosGrafica3_2($fecha_ini,$fecha_fin,$edad_ini,$edad_fin){
362         $arreglo = array();
363
364         $sql = "SELECT COUNT(*) from asistencias
365                 WHERE CAST(FECHA_INGRESO AS DATE) >= '$fecha_ini'
366                 AND CAST(FECHA_INGRESO AS DATE) <= '$fecha_fin'
367                 GROUP BY YEAR(FECHA_INGRESO) ASC, MONTH(FECHA_INGRESO)ASC";
368
369         if ($consulta = $this->conexion->conexion->query($sql)) {
370             while ($consulta_VU = mysqli_fetch_array($consulta)) {
371                 $arreglo[] = $consulta_VU;
372             }
373             return $arreglo;
374             $this->conexion->cerrar();
375         }
376     }
377
378     function prediccion_generado($id){
379         return shell_exec("python C:/xampp/htdocs/w1/pythonCode/analisis_generado.py $id");
380     }
381
382 >     function prediccion_historial($ID_test){
383         ...
384     }
385
386 >     function clasificar_datos(){
387         ...
388     }
389
390 >

```

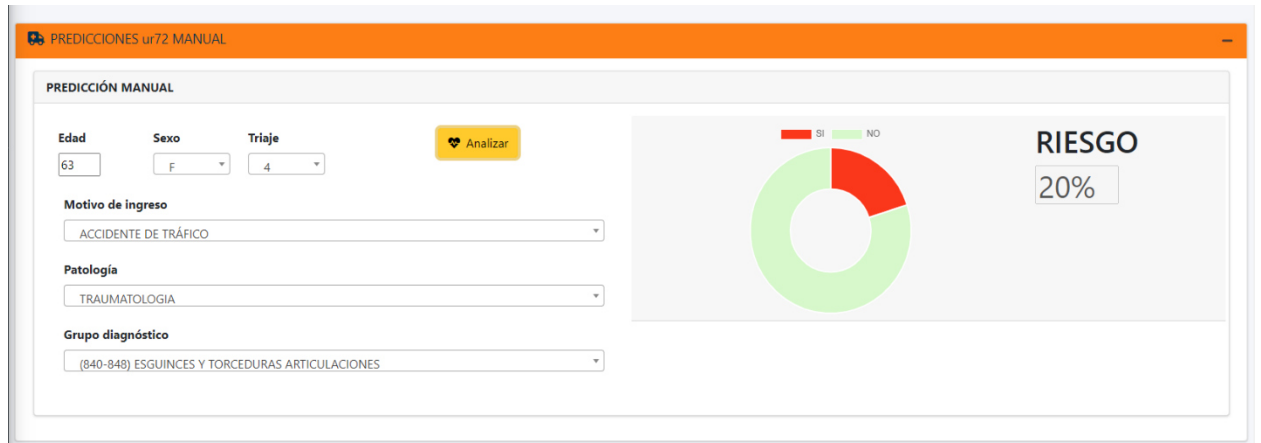
Figura 7.2.1

Después de esto, el **modelo** retorna al **controlador** la información procesada, para que este a su vez la transmita a la **vista** en el formato adecuado.

```
927 function analizar_informe_generado(){
928     var edad      = $("#paciente_edad_1").val();
929     var sexo      = $("#cbm_sexo_1").val();
930     var triaje     = $("#cbm_triaje").val();
931     var motivo    = $("#cbm_m_i").val();
932     var patologia  = $("#cbm_pt").val();
933     var grupo_diag = $("#cbm_g_d").val();
934
935     if(edad.length==0)
936         return Swal.fire("Advertencia","Rellene el campo 'EDAD'", "warning");
937
938     $.ajax({
939         "url": "../controlador/user/controlador_analizar_generado.php",
940         type: 'POST',
941         data: {
942             edad: edad,
943             sexo: sexo,
944             triaje: triaje,
945             motivo: motivo,
946             patologia: patologia,
947             grupo_diag: grupo_diag
948         }
949     }).done(function(resp){
950         /* refresh gráfica */
951         if (window.myPie_man) {
952             window.myPie_man.clear();
953             window.myPie_man.destroy();
954         }
955
956         var percent = [];
957         var result = resp.split(";");
958         percent[0] = result[1];
959         percent[1] = 100 - percent[0];
960
961         $("#porcentaje2").val(percent[0] + "%");
962         var ctx = document.getElementById('myChart_man');
963         window.myPie_man = new Chart(ctx, {
964             type: 'doughnut',
965             data: {
966                 labels: ['SI', 'NO'],
967                 datasets: [
968                     {
969                         data: percent, backgroundColor: ['#FA361B', '#D6F8CA'],
970                     }
971                 ]
972             }
973         });
974     });
975 }
```

En el ejemplo expuesto, el resultado final que retorna, es un número que indica la predicción calculada por la red neuronal. Este dato lo muestra por la vista en formato numérico y gráficamente.

Para generar el gráfico utilizamos **Chart.js**²¹, que es una biblioteca JavaScript gratuita y de código abierto para la visualización de datos, dispone de diferentes de gráficos como barras, líneas, área o doughnut que es el que se ha empleado aquí.



9 EVALUACIÓN

Para poder evaluar la eficacia de nuestra aplicación, no se disponía la posibilidad de implementarlo en un hospital y que fuera testado por el personal al que va dirigido.

Lo que si se dispone es de una gran cantidad de datos para hacer pruebas y valorar la eficacia de las predicciones. Con este fin se ha realizado un script de ensayos para comprobar la eficacia de los pronósticos.

Tenemos unos 200.000 historiales de pacientes, de los cuales el 8%, unos 16.000, son retornos al servicio de urgencias. Para entrenar la red neuronal necesitamos un numero equitativo de casos que vuelven y los que no. Bajo estos parámetros creamos un sencillos programa que realiza las siguientes tareas.

- Selecciona aleatoriamente 10.000 historiales que no han retornado y otros tantos que si lo han hecho.
- Este grupo de 20.000 lo divide en dos:
 - Grupo de entrenamiento: 5000 si y 5000 no.
 - Grupo de test: 5000 si y 5000 no.
- Se realiza el entrenamiento de la red con su grupo correspondiente.
- Se realizan las predicciones con el grupo de test.
- Al tener los resultados reales del grupo de test, se puede evaluar el porcentaje de aciertos que se ha obtenido.

Se repite el proceso 10 veces, cada una de ellas los datos escogidos son diferentes al ser la selección aleatoria. La media de aciertos obtenida ha sido del 58% (*Figura 9*), un dato que podemos considerar aceptable, teniendo en cuenta que la red puede optimizarse si se invierte tiempo en ajustar los parámetros.

Con este resultado se podría decir que, en caso de haber empleado la aplicación, casi 6 de cada 10 personas que retornaron a urgencias no hubiesen sido dadas de alta.

```
Epoch 3/10
167/167 [=====] - 0s 2ms/step - loss: 0.6872 - accuracy: 0.5687
Epoch 4/10
167/167 [=====] - 0s 2ms/step - loss: 0.6834 - accuracy: 0.5657
Epoch 5/10
167/167 [=====] - 0s 2ms/step - loss: 0.6676 - accuracy: 0.5828
Epoch 6/10
167/167 [=====] - 0s 2ms/step - loss: 0.6706 - accuracy: 0.5837
Epoch 7/10
167/167 [=====] - 0s 2ms/step - loss: 0.6803 - accuracy: 0.5800
Epoch 8/10
167/167 [=====] - 0s 2ms/step - loss: 0.6725 - accuracy: 0.5871
Epoch 9/10
167/167 [=====] - 0s 2ms/step - loss: 0.6567 - accuracy: 0.5965
Epoch 10/10
167/167 [=====] - 0s 2ms/step - loss: 0.6590 - accuracy: 0.5998
Aciertos test 10 --> 58.24 %
[59 58 59 58 58 58 58 54 55 58]
Acierto medio total 58.024 %

C:\xampp\htdocs\w1\pythonCode>
```

Figura 9

10 CONCLUSIONES Y TRABAJO FUTURO

Llega el momento de hacer balance y analizar si se ha conseguido lo pretendido, es aquí donde debo convencer al tribunal de mi buen hacer.

10.1 CONCLUSIONES

La primera idea que me transmitió mi profesor a la hora de presentarme este proyecto fue, “crear una herramienta que pronostique el riesgo de un paciente a volver al servicio de urgencias de un hospital, si es dado de alta”.

Bastante claro y conciso, por tanto, el objetivo también lo era. A partir de ese momento mis preocupaciones se centraron en la manera de conseguirlo, no había una tecnología o lenguaje de programación preestablecido para realizarlo.

Durante estos meses se ha tratado de modelar esa herramienta, si buscamos la principal funcionalidad que se buscaba en ella, la encontramos. Por tanto, en ese sentido, siento que el tiempo invertido ha obtenido sus frutos.

No obstante, no olvidemos que es una primera versión y tampoco se ha podido probar en un entorno verídico como un hospital para su evaluación. Para obtener una opinión objetiva sobre la aplicación, debemos basarnos en las pruebas teóricas realizadas, las cuales aportan un resultado aceptable.

Como ventaja de la aplicación destacaría que es sencilla, todo es fácilmente accesible y no requiere de grandes conocimientos informáticos para utilizarla. En un breve espacio de tiempo se puede conocer y dominar todas las funcionalidades.

También se podrían considerar algunos inconvenientes en ella, por ejemplo, necesita de un cargado manual de los expedientes. No se actualizan los datos de manera continua, solamente cuando alguna persona agrega nuevos historiales.

A modo de conclusión final me gustaría pensar que se ha conseguido el propósito, que, aunque es mejorable, puede afrontar el reto de presentar una predicción con un buen porcentaje de acierto.

<p> Si se puede imaginar, se puede programar </p>

10.2 CONCLUSIONS

The first idea that my teacher transmitted to me when he presenting this project to me was, "to create a tool that predicts the risk of a patient returning to the emergency department of a hospital, if he is discharged."

Quite clear and concise, therefore, the objective was also. From that moment my concerns were focused on how to achieve it, there was no technology or pre-established programming language to do it.

During these months I have tried to model this tool, if we look for the main function that was sought in it, we will find it. So, in that point, I feel that the time invested was valid.

However, we can't forget that it is a first version and it has not been tested in a true environment such as a hospital for evaluation. To obtain an objective opinion on the application, we must base ourselves on the theoretical tests carried out, which provide an acceptable result.

As an advantage of the application, I would highlight that it is simple, everything is easily accessible and does not require great computer knowledge to use it. In a short space of time, you can learn and handle all functionalities.

We could consider also some drawbacks in it, for example, it needs a manual loading of the files. The data is not updated continuously, only when someone adds new records.

As final conclusion, I would like to think that the purpose has been achieved, that, although it can be improved, it can accept the challenge of perform a prediction with a good percentage of correctness.

<p> If you can imagine it, you can program it </p>

10.3 TRABAJO FUTURO

Entre las posibles mejoras y nuevas funcionalidades futuras, aquí se citan algunas que se me ocurren desde el punto de vista del espectador de un servicio de urgencias. Si se preguntase a alguna persona que trabaja en este campo, seguramente apuntaría con más exactitud nuevas ideas o enfoques.

Como en mi caso no conozco ninguna persona relacionada con este trabajo, mis propuestas pueden ser algo más generalizadas, así pues, se debería contrastar este trabajo con personal especializado si se decidiese evolucionar.

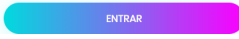
Posibles mejoras:

- **Predicciones:** actualmente la configuración de la red neuronal para el usuario es fija, es decir, no tiene opción de hacer ningún cambio, número de capas, neuronas, entrenamiento. Podría añadirse una pestaña donde algunos usuarios, no todos, pudiesen variar los parámetros con el fin de mejorar los resultados.
- **Listado de historiales:** cada historial cuenta con 62 campos, en la aplicación cuando se despliega el listado, solo se muestran algunos, los que se han creído más importantes. Sería bueno que el usuario pudiese seleccionar los campos que quiere que se muestren, a cada usuario, dependiendo de su especialidad le pueden interesar diferentes informaciones.
- **Gráficas:** se cuenta con una inmensa cantidad de información en la base de datos, la aplicación muestra 3 gráficas diferentes a modo de ejemplo. En este aspecto hay un gran número de posibles gráficas a añadir.
- **Notificaciones:** podría ser interesante dotar de una funcionalidad de notificaciones. Un apartado donde recoja aspectos relacionados con la aplicación si hay modificaciones o mejoras.
- **AYUDA:** en la parte inferior izquierda se ha colocado una zona “AYUDA” donde poder obtener información o consejos sobre la aplicación. Por falta de tiempo, no se ha podido completar y no muestran ningún texto.

11 GUÍA DE USO

El objetivo de esta sección es mostrar al usuario tanto las funcionalidades de la aplicación como la manera de utilizarlas, de una manera amena y sencilla.

11.1 INICIO

La primera pantalla que no encontramos al acceder es la de registro, en ella deberá introducir el nombre de usuario y la contraseña en los apartados correspondientes (*Figura 11.1.1*). Una vez introducidos al pulsar sobre el botón , pueden ocurrir tres cosas:

1. Los datos son correctos, es informado y accede a la aplicación (*Figura 11.1.2*).
2. Los datos son incorrectos, es informado y permanece en la página inicial (*Figura 11.1.3*).
3. Los datos son correctos, pero es un usuario que no se encuentra “activo”, esta cualidad se explica más adelante, pero básicamente, es una persona que está dada de alta en la base de datos, pero por motivos estimados por el administrador se encuentra deshabilitado temporalmente para utilizar la herramienta. En este caso también es informado y permanece en la página inicial (*Figura 11.1.4*).

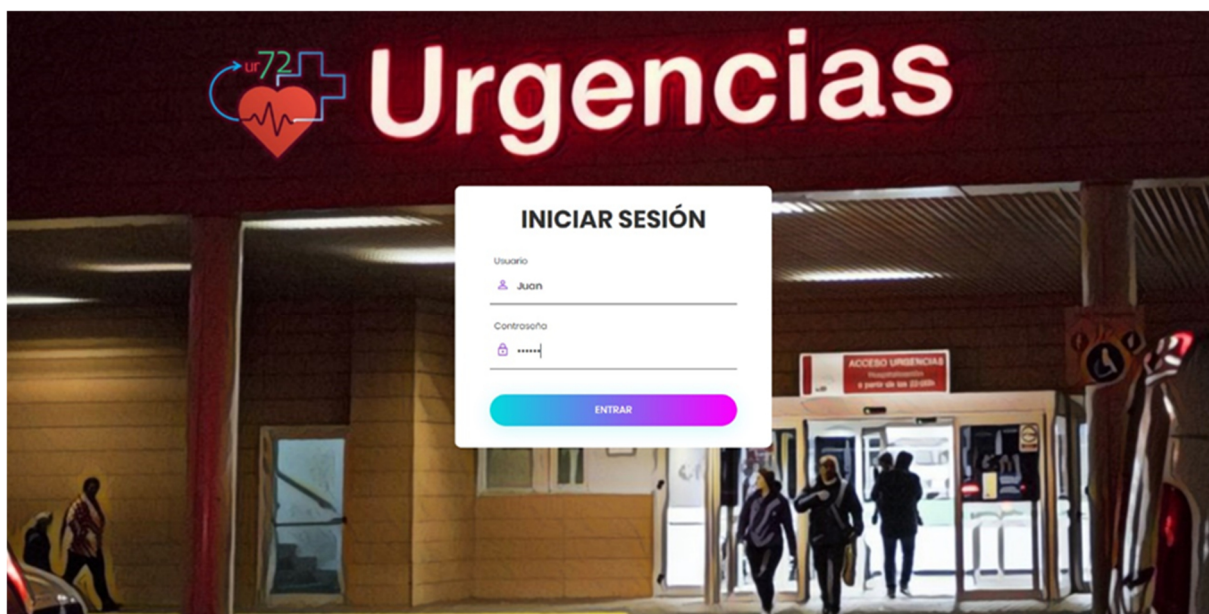


Figura 11.1.1



Figura 11.1.2

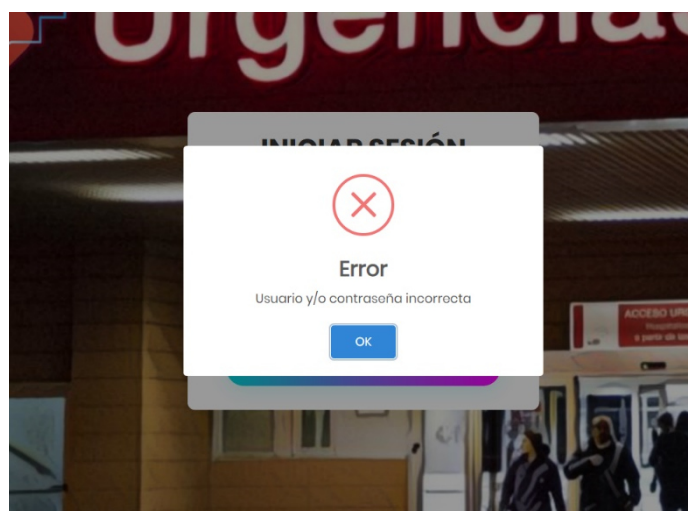


Figura 11.1.3

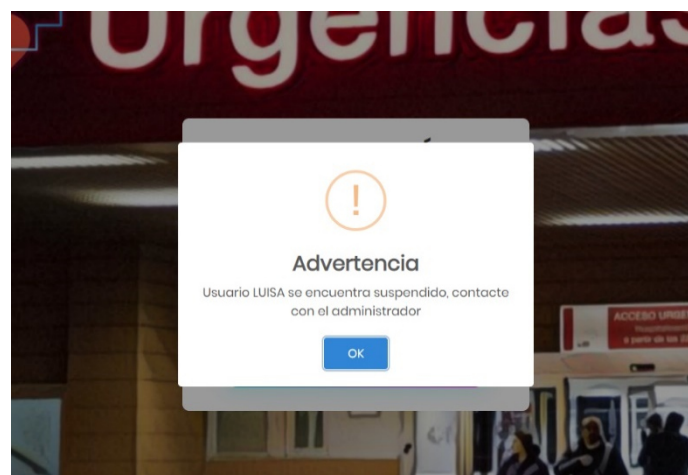
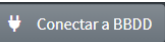


Figura 11.1.4

Una vez nos hayamos registrado correctamente nos redirecciona automáticamente a la página principal (*Figura 11.1.5*).



Figura 11.1.5

Para poder utilizar la aplicación, lo primero que tenemos que hacer es pulsar el botón de la parte superior para conectar con la base de datos . En caso de no hacerlo al desplegar cualquier pestaña (pulsar “+” a la derecha de cada una de ellas), se nos recordará que lo hagamos (*Figura 11.1.6*).

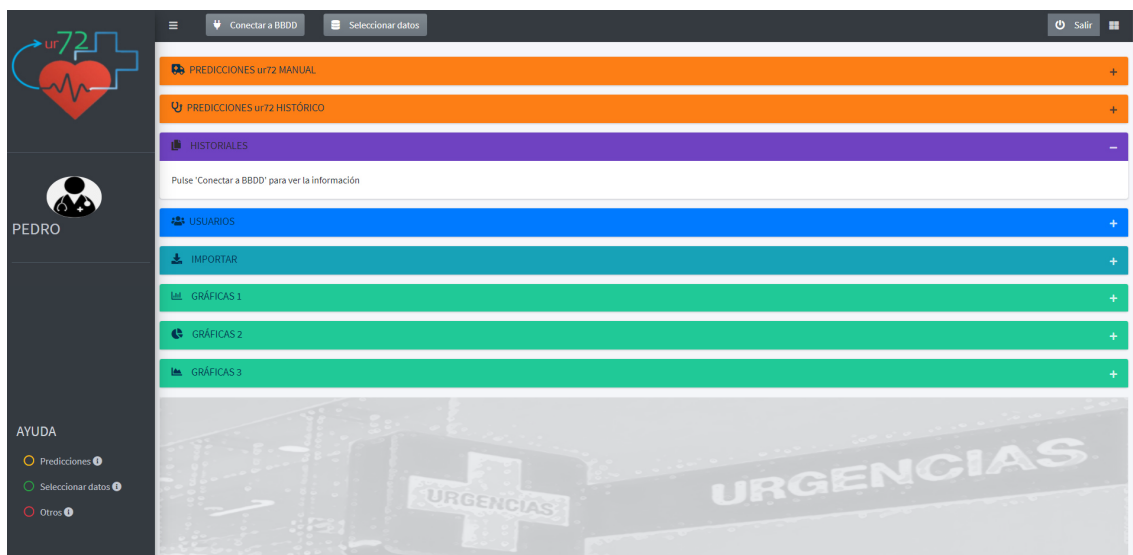


Figura 11.1.6

11.2 IMPORTAR DATOS

Si ya estamos conectados a la base de datos ya podremos interactuar con la aplicación. Pero es posible que sea la primera vez que se accede y aún no se dispongan de registros en la base de datos. Si es este el caso, deberemos incorporar los informes a nuestra base, para ello utilizamos la herramienta “IMPORTAR” (Figura 11.2.1).

Al pinchar en “Seleccionar archivo” nos abrirá el explorador para seleccionar los archivos donde se registran las asistencias de pacientes (Figura 10.8). Importante, el archivo debe ser de formato tipo Excel, en caso contrario aparecerá un mensaje de error.

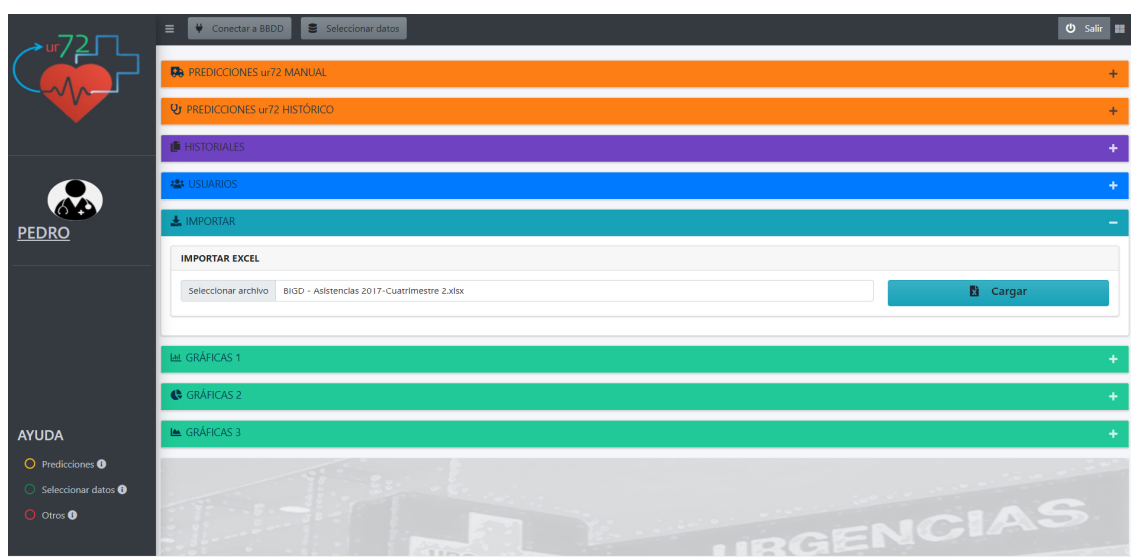
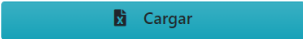


Figura 11.2.1

Después de seleccionar el archivo pulsaremos el botón , se iniciará la importación de datos y una vez terminada se indicarán el número de registros incorporados (Figura 11.2.2). Solo incorporará historiales que no estén en la base.

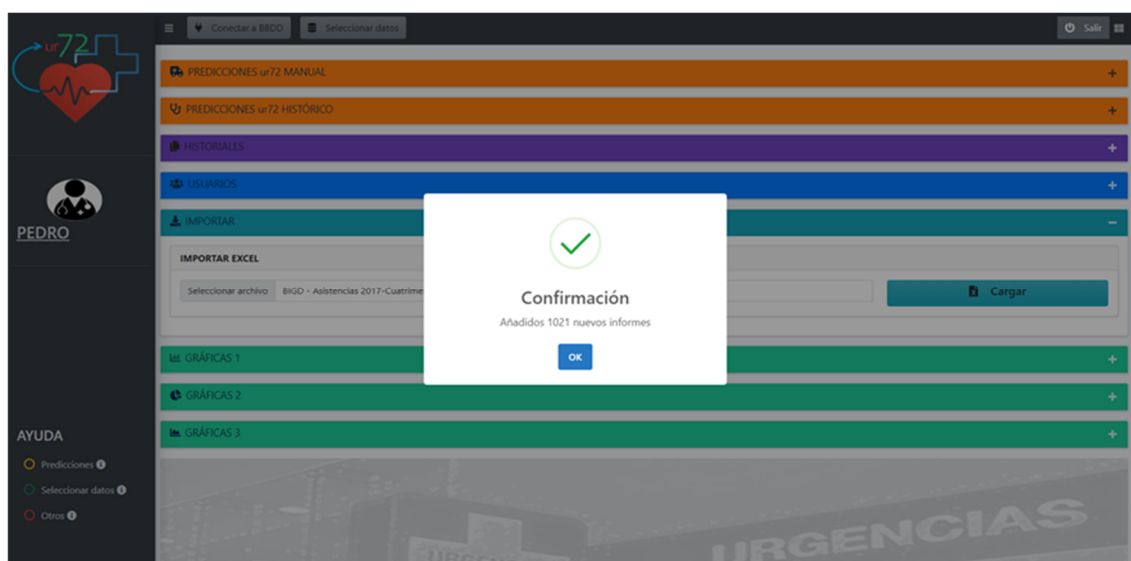



Figura 11.2.2

11.3 HISTORIALES

AL disponer de datos, se podrán visionar en el apartado “HISTORIALES” (Figura 11.3). En el listado podremos organizar la disposición de los mismos en función del campo que deseemos, pinchando en  se organizará de mayor a menor o viceversa.

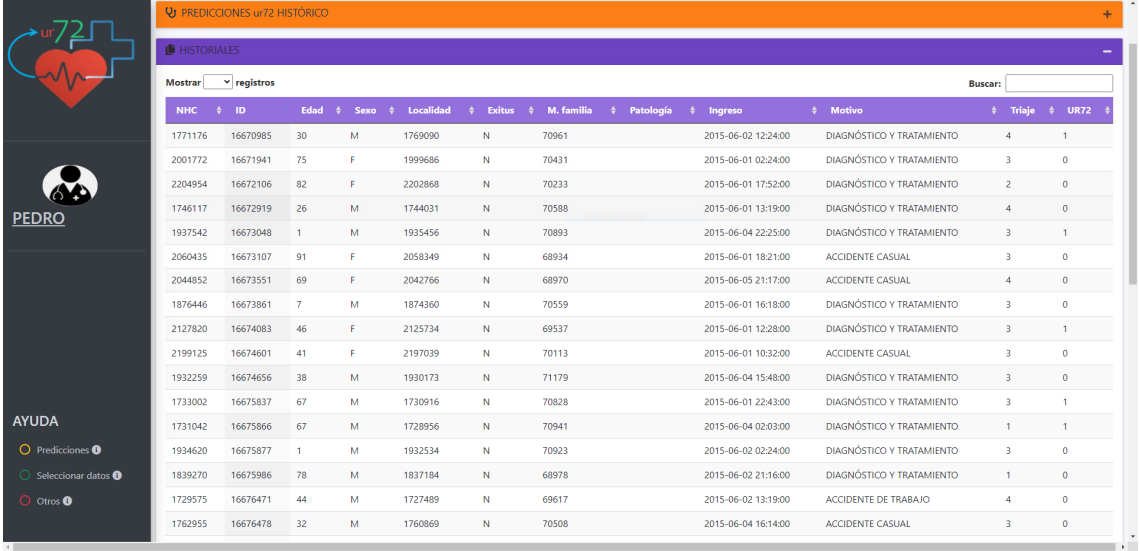


Figura 11.3 muestra la interfaz de usuario de la aplicación, específicamente la sección de "HISTORIALES". El panel principal muestra una lista de registros de pacientes con los siguientes campos: NHC, ID, Edad, Sexo, Localidad, Exitus, M. familia, Patología, Ingreso, Motivo, Triaje y UR72. La lista contiene 20 registros de pacientes, cada uno con sus respectivos datos personales y médicos.

NHC	ID	Edad	Sexo	Localidad	Exitus	M. familia	Patología	Ingreso	Motivo	Triaje	UR72
1771176	16670985	30	M	1769090	N	70961		2015-06-02 12:24:00	DIAGNÓSTICO Y TRATAMIENTO	4	1
2001772	16671941	75	F	1999686	N	70431		2015-06-01 02:24:00	DIAGNÓSTICO Y TRATAMIENTO	3	0
2204954	16672106	82	F	2202868	N	70233		2015-06-01 17:52:00	DIAGNÓSTICO Y TRATAMIENTO	2	0
1746117	16672919	26	M	1744031	N	70588		2015-06-01 13:19:00	DIAGNÓSTICO Y TRATAMIENTO	4	0
1937542	16673048	1	M	1935456	N	70893		2015-06-04 22:25:00	DIAGNÓSTICO Y TRATAMIENTO	3	1
2060435	16673107	91	F	2058349	N	68934		2015-06-01 18:21:00	ACCIDENTE CASUAL	3	0
2044852	16673551	69	F	2042766	N	68970		2015-06-05 21:17:00	ACCIDENTE CASUAL	4	0
1876446	16673861	7	M	1874360	N	70559		2015-06-01 16:18:00	DIAGNÓSTICO Y TRATAMIENTO	3	0
2127820	16674083	46	F	2125734	N	69537		2015-06-01 12:28:00	DIAGNÓSTICO Y TRATAMIENTO	3	1
2199125	16674601	41	F	2197039	N	70113		2015-06-01 10:32:00	ACCIDENTE CASUAL	3	0
1932259	16674656	38	M	1930173	N	71179		2015-06-04 15:48:00	DIAGNÓSTICO Y TRATAMIENTO	3	0
1733002	16675837	67	M	1730916	N	70828		2015-06-01 22:43:00	DIAGNÓSTICO Y TRATAMIENTO	3	1
1731042	16675866	67	M	1728956	N	70941		2015-06-04 02:03:00	DIAGNÓSTICO Y TRATAMIENTO	1	1
1934620	16675877	1	M	1932534	N	70923		2015-06-02 02:24:00	DIAGNÓSTICO Y TRATAMIENTO	3	0
1839270	16675986	78	M	1837184	N	68978		2015-06-02 21:16:00	DIAGNÓSTICO Y TRATAMIENTO	1	0
1729575	16676471	44	M	1727489	N	69617		2015-06-02 13:19:00	ACCIDENTE DE TRABAJO	4	0
1762955	16676478	32	M	1760869	N	70508		2015-06-04 16:14:00	ACCIDENTE CASUAL	3	0

Figura 11.3

11.4 USUARIOS

Esta sección está orientada a la gestión del personal que puede utilizar la aplicación, solo se mostrará a las personas que estén registradas como *administrador* (Figura 11.4.1).

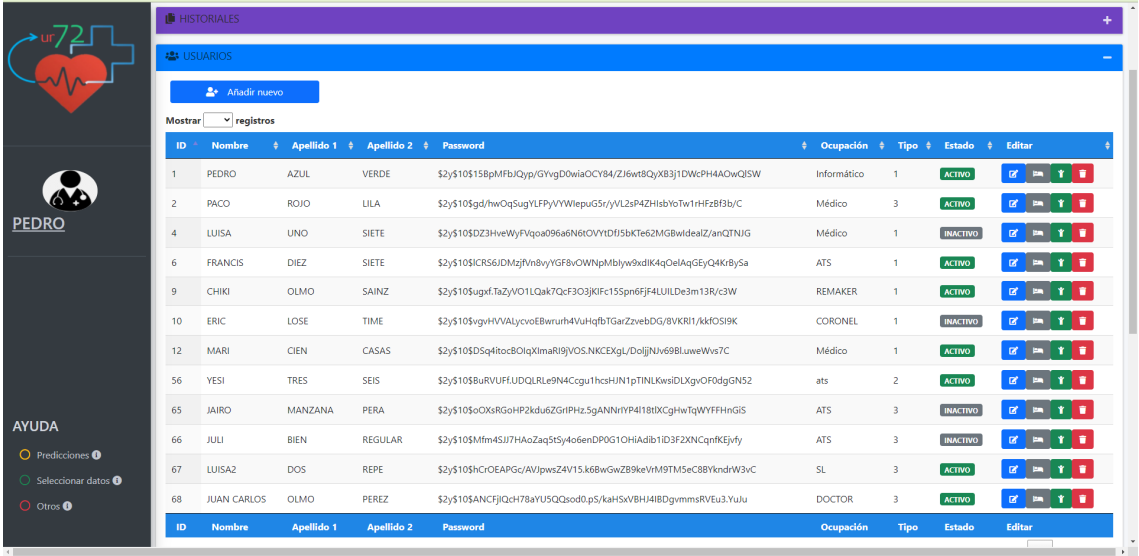


Figura 11.4.1 muestra la interfaz de usuario de la aplicación, específicamente la sección de "USUARIOS". El panel principal muestra una lista de usuarios registrados con los siguientes campos: ID, Nombre, Apellido 1, Apellido 2, Password, Ocupación, Tipo, Estado y Editar. La lista contiene 16 usuarios, cada uno con sus respectivos datos personales y laborales.

























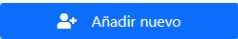

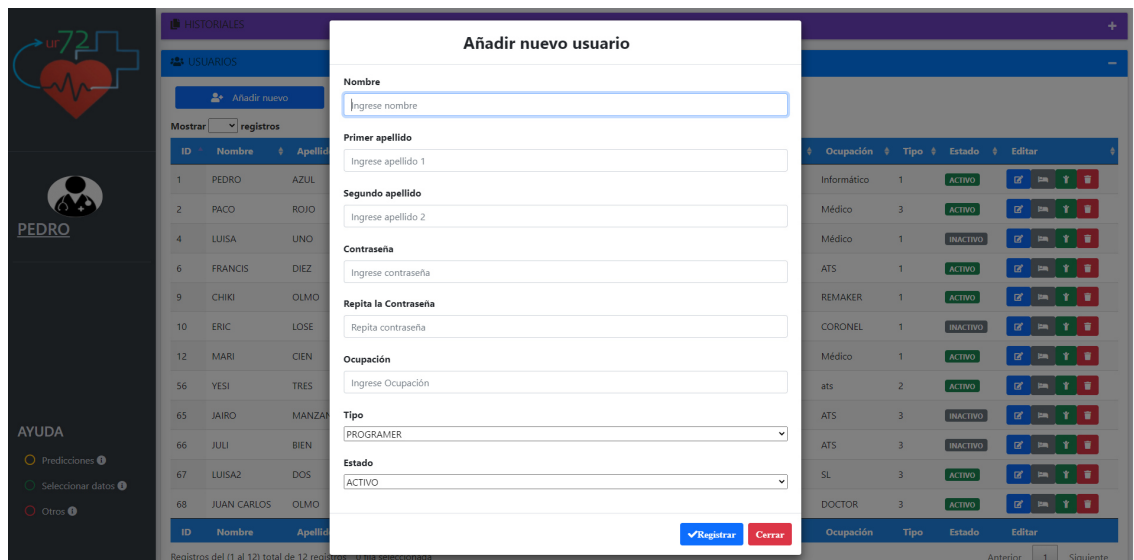
ID	Nombre	Apellido 1	Apellido 2	Password	Ocupación	Tipo	Estado	Editar
1	PEDRO	AZUL	VERDE	\$2y\$10\$158pMfBjQyp/GYvD0wiaOCY84/ZJ6wt8QyXB3j1DWcPH4AOwQISW	Informático	1	ACTIVO	 
2	PAICO	ROJO	LILA	\$2y\$10\$gd/hwQo5ugYLFpy/VWlepuG5r/yL2sP4ZIsbYoTw1HfzB3b/C	Médico	3	ACTIVO	 
4	LUISA	UNO	SIETE	\$2y\$10\$DZ3HveWfYfQoa96a6N6OVYDf15bKTe62MGBwdeazZ/anC7N1G	Médico	1	INACTIVO	 
6	FRANCIS	DIEZ	SIETE	\$2y\$10\$ICRS6IDMzj/Vn8vyVGF8vOWNpMblyw9dKlKqOelAqGEyQ4KcBySa	ATS	1	ACTIVO	 
9	CHIKI	OLMO	SAINZ	\$2y\$10\$ugxfTaZyVO1LQak7QcF3O3JfKf155pn6FJ4LULDe3m13R/c3W	REMAKER	1	ACTIVO	 
10	ERIC	LOSE	TIME	\$2y\$10\$vgvRvVALycoEBwruh4VuHqtbTGarZzevDG/8VKR1t/kkOSI9K	CORONEL	1	INACTIVO	 
12	MARI	CIEEN	CASAS	\$2y\$10\$D5q4t0cB0lqXmarI9jVOS.NKCEXg/LDoljJNuv698LuwewWvs7C	Médico	1	ACTIVO	 
56	YESI	TRES	SEIS	\$2y\$10\$BuRVUfF.UOQLRLe9N4CcgulhcsHJN1pTINLkwsIDLXgvOFodgGN52	ats	2	ACTIVO	 
65	JAIRO	MANZANA	PERA	\$2y\$10\$oxRGoHP2kdu6ZGrPHz.5gANNrYP4l18tDCgHwTqWYFFHnGIS	ATS	3	INACTIVO	 
66	JULI	BIEN	REGULAR	\$2y\$10\$fm4Sj7HAozQa5S5y406enDPOG1OH1Adib1D3FZKNcqnKCEjvly	ATS	3	INACTIVO	 
67	LUISA2	DOS	REPE	\$2y\$10\$HcROEAPGc/AVJpwsZ4V15.k68wGwZB9keVrM9TM5eC8BYkndrW3vC	SL	3	ACTIVO	 
68	JUAN CARLOS	OLMO	PEREZ	\$2y\$10\$ANCFJQch78ayUSQ2sod0pS/kaHsxvBHJ4IBDgymmsRVeu3.Yulu	DOCTOR	3	ACTIVO	 

Figura 11.4.1

Pulsando sobre  podremos añadir nuevos usuarios, se desplegará un formulario en donde habrá que completar todos los campos (*Figura 11.4.2*). Una vez rellenos, pulsamos , si hubiese algún error nos lo indica, si está todo correcto, añade al usuario a la base de datos y al listado.







The screenshot shows a web application interface. On the left is a sidebar with a user profile for 'PEDRO' and a menu with 'AYUDA', 'Predicciones', 'Seleccionar datos', and 'Otros'. The main area displays a table of users. Overlaid on this is a modal window titled 'Añadir nuevo usuario'. The modal contains the following fields:

- Nombre:** Ingrese nombre
- Primer apellido:** Ingrese apellido 1
- Segundo apellido:** Ingrese apellido 2
- Contraseña:** Ingrese contraseña
- Repita la Contraseña:** Repita contraseña
- Ocupación:** Ingrese Ocupación
- Tipo:** PROGRAMER (dropdown menu)
- Estado:** ACTIVO (dropdown menu)

At the bottom of the modal are two buttons: 'Registrar' (with a checkmark icon) and 'Cerrar' (in red).

Figura 11.4.2

Otras de las acciones que pueden realizar en este apartado son:

-  **Editar usuario:** para modificar datos del usuario.
-  **Desactivar usuario:** desactiva al usuario, mientras esté en este estado no podrá acceder a la aplicación.
-  **Activar usuario:** pone al usuario en modo activo. Puede acceder a la aplicación.
-  **Borrar usuario:** borra a un usuario de la base de datos.

En las operaciones **Desactivar**, **Activar** y **Borrar**, se pedirá confirmación mediante una ventana emergente. (*Figura 11.4.3*)

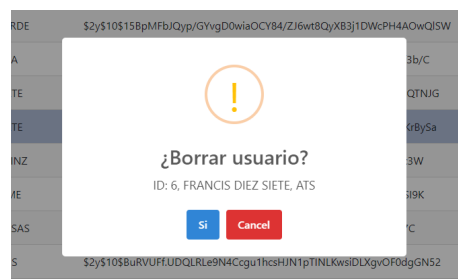


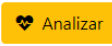
Figura 11.4.3

11.5 PREDICCIONES ur72 MANUAL

En este apartado se simula el historial de un paciente y se analiza el riesgo de retorno que tendría. Simplemente hay que agregar el dato “Edad” y seleccionar de los desplegables las opciones que deseen analizar. (Figura 11.5.1)

The screenshot shows the 'PREDICCIONES ur72 MANUAL' interface. On the left is a dark sidebar with a logo, the name 'PEDRO', and an 'AYUDA' section with links to 'Predicciones', 'Seleccionar datos', and 'Otros'. The main area has an orange header. Below it, the 'PREDICCIÓN MANUAL' section contains input fields for 'Edad' (with a value of 18), 'Sexo' (M), and 'Triaje' (1), followed by a yellow 'Analizar' button. Below these are three dropdown menus: 'Motivo de ingreso' (INICIATIVA PROPIA), 'Patología' (MEDICINA GENERAL), and 'Grupo diagnóstico' (001-009 ENFERMEDADES INFECCIOSAS INTESTINALES). To the right, a large box labeled 'RIESGO' is empty. At the bottom, there is a horizontal menu with options: 'PREDICCIONES ur72 HISTÓRICO', 'HISTORIALES', 'USUARIOS', 'IMPORTAR', 'GRÁFICAS 1', and 'GRÁFICAS 2'.

Figura 11.5.1

A continuación, se pulsa  y en la parte derecha se muestra el resultado. (Figura 11.5.2)

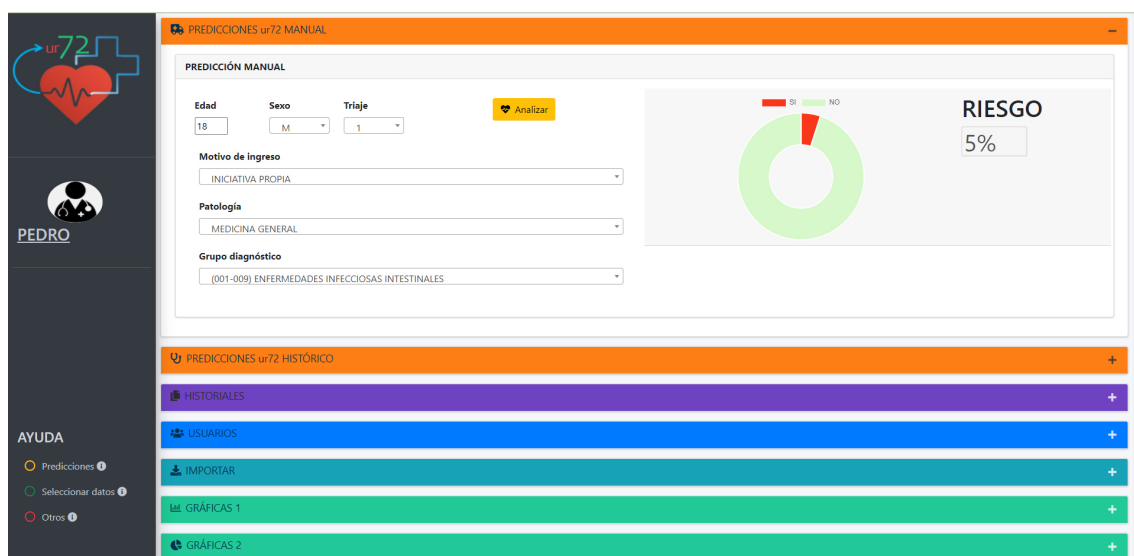


Figura 11.5.2

11.6 PREDICCIONES ur72 HISTÓRICO

La funcionalidad es la misma que la del apartado anterior, obtener una predicción, la diferencia radica en que en este caso la información se obtiene de un informe de la base de datos.

Para obtenerlos, debemos introducir en la casilla “ID” el número de informe que queremos evaluar (Figura 11.6.1), tras ello pulsamos **Buscar** y se cargarán los datos del informe (Figura 11.6.2). A continuación, se pulsa **Analizar** y en la parte derecha se muestra el resultado.

PREDICCIONES ur72 HISTÓRICO

PREDICCIÓN HISTÓRICO

ID 17245016 Buscar Analizar

RIESGO

Fecha Paciente(NHC) Edad Sexo Triaje R72

Patología Motivo de ingreso

Diagnóstico principal Grupo diagnóstico

Procesos Destino

Comentarios Juicio clínico

Figura 11.6.1

PREDICCIONES ur72 HISTÓRICO

PREDICCIÓN HISTÓRICO

ID 17245016 Buscar Analizar

RIESGO

Fecha Paciente(NHC) Edad Sexo Triaje R72

Patología Motivo de ingreso

Diagnóstico principal Grupo diagnóstico

Procesos Destino

Comentarios Juicio clínico

Figura 11.6.2


11.7 GRÁFICAS

Existen tres pestañas destinadas a la representación gráfica de los datos que gestionamos.

11.7.1 GRÁFICAS 1

Muestra en una gráfica de barras la cantidad de ingresos por edades (*Figura 11.7.1*). Si se desea se pueden establecer los siguientes parámetros para acotar los datos:

- EDAD: entre 0 y 110 años.
- FECHA: del día inicial mm/dd/aaaa al día final mm/dd/aaaa.
- MOTIVO DE INGRESO.
- PATOLOGÍA.
- GRUPO DIAGNÓSTICO.

Al pulsar  , se genera la gráfica bajo los parámetros establecidos.

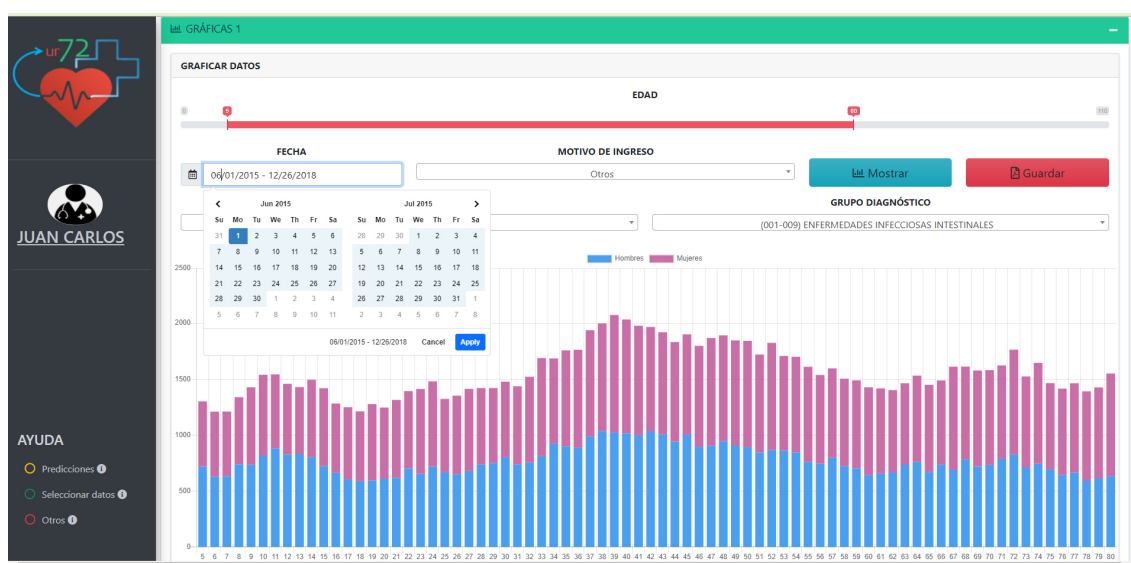
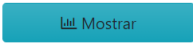


Figura 11.7.1

11.7.2 GRÁFICAS 2

Muestra dos gráficas en formato “pie” con los números de ingresos, una los clasifica por PATOLOGÍA y la otra por MOTIVO DE INGRESO (*Figura 11.7.2*). Si se desea se pueden establecer los siguientes parámetros para acotar los datos:

- EDAD: entre 0 y 110 años.
- FECHA: del día inicial mm/dd/aaaa al día final mm/dd/aaaa.

Al pulsar , se genera la gráfica bajo los parámetros establecidos.

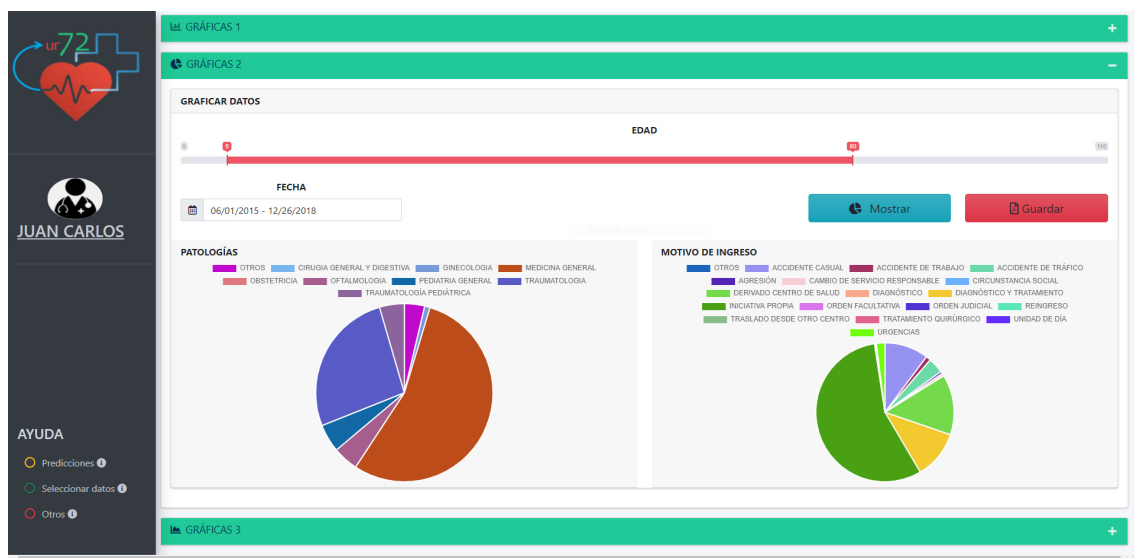
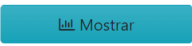


Figura 11.7.2

11.7.3 GRAFICAS 3

Muestra en una gráfica lineal el número de ingresos y retornos mensuales. (Figura 11.7.3). Si se desea se pueden establecer los siguientes parámetros para acotar los datos:

- EDAD: entre 0 y 110 años.
- FECHA: del día inicial mm/dd/aaaa al día final mm/dd/aaaa.
- MOTIVO DE INGRESO.
- PATOLOGÍA.
- GRUPO DIAGNÓSTICO.

Al pulsar , se genera la gráfica bajo los parámetros establecidos.

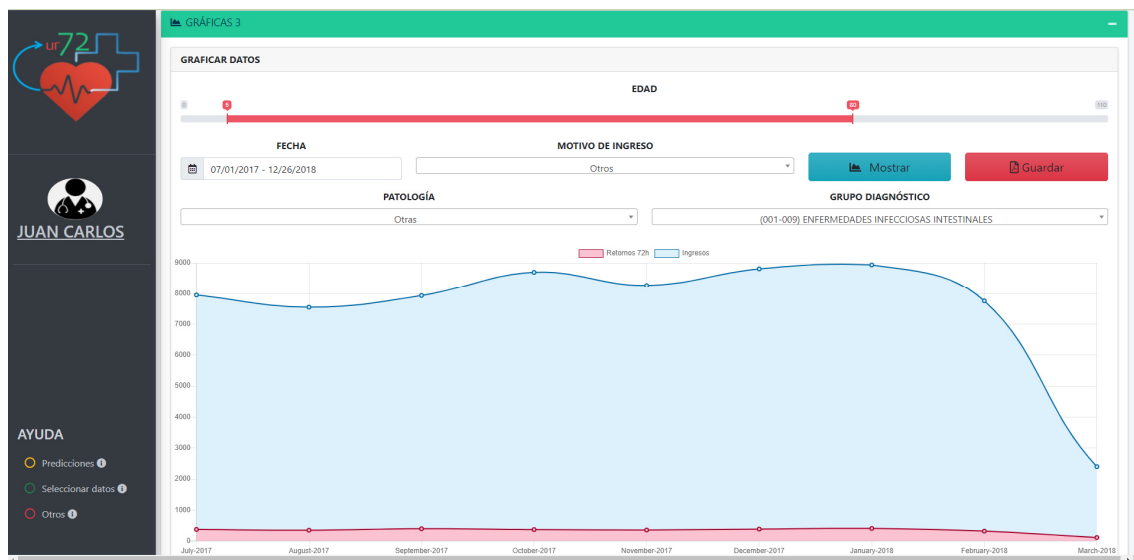

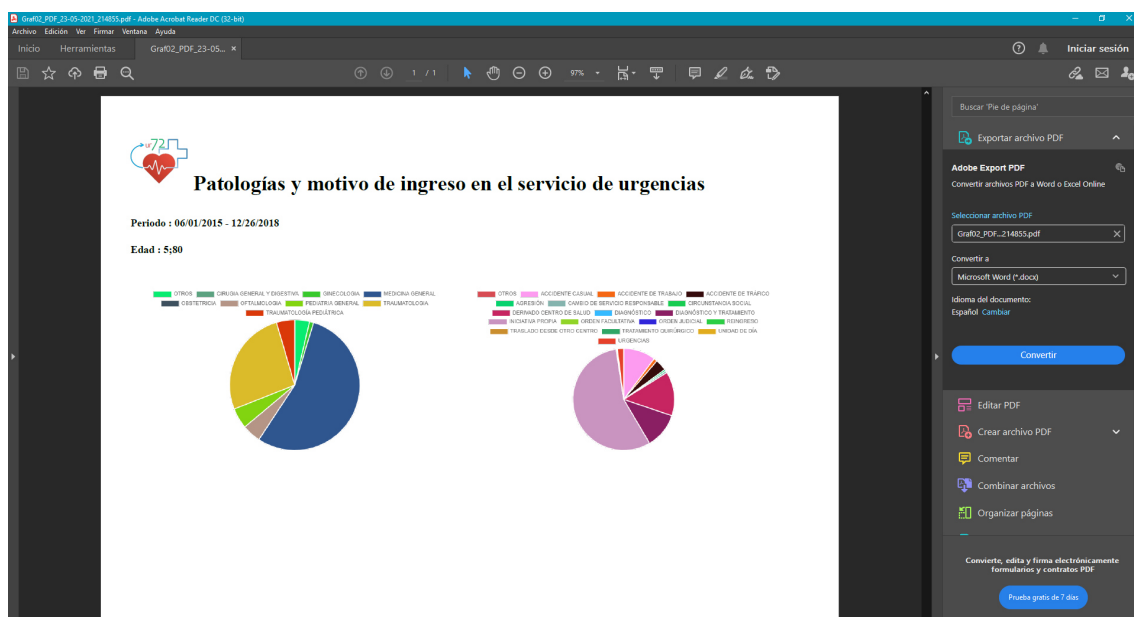
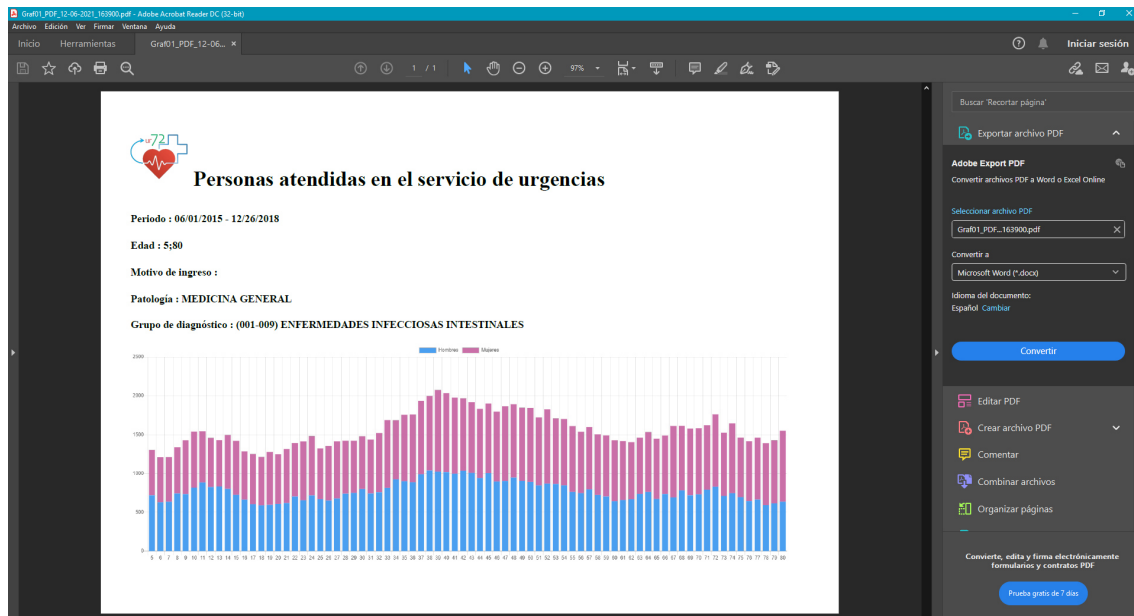


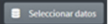
Figura 11.7.3

11.7.4 GUARDAR PDF

En cada una de las gráficas de los apartados anteriores, existe la posibilidad de guardar el resultado de las mismas. Pulsando  Guardar se obtendrá un archivo en formato PDF, nombrado con el número, fecha y hora de la gráfica.



11.8 SELECCIONAR DATOS

En la parte superior de la pantalla, se encuentra el botón , la funcionalidad que tiene es la de renovar los datos de entrenamiento de la red neuronal.

Para ajustar la red se utilizan aproximadamente un 10% de los historiales, de los cuales la mitad son pacientes que han vuelto y la otra mitad que no lo hicieron. Es conveniente por pulsar el botón de vez en cuando, al hacerlo, se formará un grupo aleatorio nuevo de datos de entrenamiento.



12 REPOSITORIO DE CÓDIGO

El código realizado se encuentra disponible en la siguiente ubicación:

https://drive.google.com/drive/folders/1R3bONKoBj4L6h1_9sbW3sFDdRRxbttfX?usp=sharing

13 BIBLIOGRAFÍA

Redes neuronales

- <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model>
- <https://blogs.imf-formacion.com/blog/tecnologia/crear-red-neuronal-en-phyton-202009/>
- <https://www.elsevier.es/es-revista-atencion-primaria-27-articulo-redes-neuronales-concepto-aplicaciones-utilidad-13033737>

Python

- <https://docs.python.org/es/3/tutorial/>
- <https://numpy.org/doc/stable/contents.html>
- <https://www.tensorflow.org/?hl=es-419>
- <https://keras.io/api/>

XAMPP

- <https://www.mclibre.org/consultar/php/otros/xampp-instalacion-windows.html>
- <https://www.mclibre.org/consultar/php/otros/xampp.html>

PHP y JS

- <https://www.php.net/manual/es/index.php>
- https://www.w3schools.com/php/php_mysql_select.asp
- <https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>

SAVANA

- <https://www.cambio16.com/las-maquinas-han-aprendido-a-analizar-casos-no-solo-la-reglas/>

13.1 REFERENCIAS

-
- ¹ **SAVANA:** <https://savanamed.com/>
 - ² **Google Drive:** <https://www.google.com/intl/es-es/drive/>
 - ³ **Visual Studio Code:** <https://code.visualstudio.com/>
 - ⁴ **XAMPP:** <https://www.apachefriends.org/es/index.html>
 - ⁵ **Apache:** https://es.wikipedia.org/wiki/Servidor_HTTP_Apache
 - ⁶ **MariaDB:** <https://es.wikipedia.org/wiki/MariaDB>
 - ⁷ **PHP:** <https://www.php.net/>
 - ⁸ **HTML5:** <https://es.wikipedia.org/wiki/HTML5>
 - ⁹ **CSS3:** <https://lenguajecss.com/css/>
 - ¹⁰ **JavaScript:** <https://lenguajejs.com/javascript/>
 - ¹¹ **AdminLTE:** <https://adminlte.io/>
 - ¹² **Bootstrap:** <https://getbootstrap.com/>
 - ¹³ **Python:** <https://www.python.org/>
 - ¹⁴ **MySQL:** <https://es.wikipedia.org/wiki/MySQL>
 - ¹⁵ **phpMyAdmin:** <https://www.phpmyadmin.net/>
 - ¹⁶ **TensorFlow:** <https://www.tensorflow.org/?hl=es-419>
 - ¹⁷ **Keras:** <https://keras.io/>
 - ¹⁸ **Hot Encoding:** <https://es.wikipedia.org/wiki/One-hot>
 - ¹⁹ **MVC:** <https://es.wikipedia.org/wiki/Modelo%E2%80%93vista%E2%80%93controlador>
 - ²⁰ **AJAX:** https://www.w3schools.com/php/php_ajax_intro.asp
 - ²¹ **Chart.js:** <https://www.chartjs.org/>